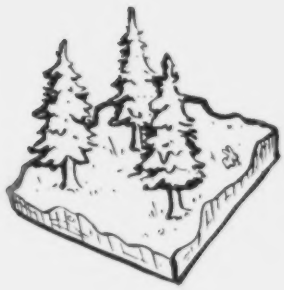
Ministry of Forests

Silviculture Information Access

Training Manual

November 2001

105-06327/



Silviculture Information Access

Training Manual

This new SIA Training Manual reflects the new information from the recent releases of ISIS, and the incorporation of the Major License Information System into the business process.

Use this manual to access the information provided by ISIS to build and generate reports for particular forest districts. Please recognize that the MLS SIA tables are no longer available, but are now amalgamated into one set of tables in a Microsoft Access 8.0 database.

Produced By

BC Ministry of Forests
Information Management Group

Updated—November 2001

2/3/4/162

Table of Contents

Table of Contents **5**

Course Overview **8**
- Intended Audience 8
- Purpose 8
- Prerequisites 8

About This Manual **9**
- Manual Layout 9
- Conventions Used in This Manual 12

SIA Background **13**
- Introduction 13
- What is Information Access? 14
- Differences Between SIA and ISIS Flexible Reports (ISR) 15

Relational Database Design **17**
- Introduction 17
- Database Management System 17
- Database Terminology 18
- Surrogate Keys 20
- Linking Tables in a Relational Database 20
- Table Relationship Diagrams 22
- Types of Relationships 23
- Mandatory vs Optional 24
- Query 24

Setting Up Your PC **25**
- Introduction 25
- Creating your SIA Database 25

The SIA Tables **31**
- Introduction 31
- SIL - Opening Table 31
- ISS - Activity Treatment Unit Table 34
- ISS - Forest Cover Table 36
- ISS - Stocking Standard Table 39
- ISS - Ecostrata Table 41
- Other Tables 43

Sample Queries **47**
- Introduction 47
- Query Elements 47
- Dynasets or Result Sets 48

Query Types....48
The Query Design Screen....49
The Query Design Toolbar....50
Using the Sample Queries....52
One-Table Queries....52
Modifying the Sample One-Table Queries....53
Sorting Records....53
Creating Sort Criteria....54
Selecting Records....57
Creating Record Selection Criteria....59
Multi-Table Queries....63
Modifying the Sample Multi-Table Queries....65

Creating One-Table Queries....67
Introduction....67
Adding Calculated Fields to a Query....71
System Generated Fields....73
Field Properties....75
Parameter Queries....79

Summary Queries....81
Creating a Summary Query....81
Cross Tabulations....87

Creating Multi-Table Queries....89
Introduction....89
Creating a Two Table Query....90
Joining Tables....93
More Complex Joins Between Tables....94

Make Table Queries....97
Warnings on Make-Table Queries....97
Creating a Make-Table Query....98

Query on Query....101
Introduction....101
Creating a Query on a Query....102
Summarizing Data Using a Query on Query....105

Advanced Queries....107
Understanding ISIS Relationships - Fundamental Rules....107
Distinguishing Between Different Business Areas....109
Common Data fields with different meanings....110
Breaking down the question to build the query....111
Understanding the evolution of data requirements and the impact on reporting....115

Business Applications of SIA....117
Emerging Uses of SIA to meet other business areas needs....117

Reconciling FTAS and ISIS Information....117
Exception Reports or Locating Missing Information....118
Routine District Monitoring and Compliance Reporting....119
FS922....120
Resource Management Plans-Guidelines for the Sustainable Harvest Component....120
TSR-Regen Assumptions for managed stands....121

Module Building....123
Complex functions....124

Managing Queries....127

Reports....129
Introduction....129
Creating Reports....129

On Your Own....137
Introduction....137

Troubleshooting, Tips and Support....139
Troubleshooting....139
Tips and Tricks....141
Support....142

Appendix A....143
List of Canned SIA Queries & description....143

Appendix B....147
Module/Function examples....147
Module ConvertSpecies....150
Module FS922-ConvertActivityCodes....152

Appendix C....155
Basic Comparison Operators....155
Additional Comparison Operators....156
Building Expressions....157
Summary Operators....159

Appendix D....161
Toolbars....161
The Datasheet Screen Toolbar....161

Appendix E....163
Answer Key for Exercises....163

Course Overview

The Silviculture Information Access (SIA) database is an application that allows you to create your own queries and reports on district silviculture information using Microsoft (MS) Access. This course assumes that Access version 8.0 (also known as Access 97) is installed on your workstation.

Intended Audience

This course is intended for district personnel who are familiar with MS Access and are familiar with the information stored in the Integrated Silviculture Information System (ISIS). This course (and the SIA database) is intended for users who have specialised reporting requirements, which are not met by the mainframe reporting systems.

Purpose

The purpose of this course is to provide a detailed understanding of the SIA database and to teach querying and reporting procedures in MS Access using the SIA database.

Prerequisites

Prerequisites for this course include MS Access training, recent ISIS training or regular use of ISIS, a good understanding of database concepts, and a good understanding of silviculture data structures. It is not intended for novice users who do not have the above listed prerequisites.

About This Manual

Manual Layout

In this manual, each section will begin with an overview of the basic concepts, followed by a series of exercises designed to demonstrate basic procedures. Depending on the topic covered, a section may contain the following parts:

Alternative Methods of Performing a Command

In Windows applications there are usually two or three ways to perform most commands. This allows you to use the application in a way that is most comfortable for you. Generally, this manual will illustrate the different methods of performing a command. For example, a command can normally be executed by choosing from a menu, by clicking a button on a toolbar, by right clicking to display a context-sensitive menu, or by using a keyboard shortcut.
For example, to run a query you can do any one of the following:

Menu: **View|Datasheet View**
Query|Run

Toolbar:

Generic Procedures and Exercises

Some sections will contain a step by step procedure for performing a task, and most sections will end with an exercise. Step-by-step procedures are given for reference, and the exercises are designed to be completed in class using the SIATRAIN.MDB, which your instructor will supply.

Procedures are listed as a series of **lettered** steps. For example:

To modify the properties of a field:

a) Click to place the insert point in the field cell then do any one of the following:

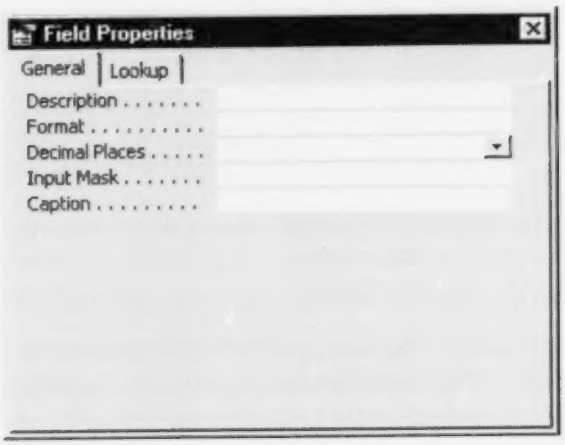
Choose **View|Properties**

Click the *Properties* button on the toolbar

Right-click in the *Field* cell and choose *Properties* from the short-cut menu

b) Access will display the *Field Properties* dialog box (shown below).

c) Customize the required properties, then press <Alt><F4> or click the close button to return to the query grid.



***Note**: The Format and Decimal Places text boxes provide pop-up buttons which allow you to select from a list of available options for the particular property.*

Exercises are listed as a series of **numbered** steps. For example:

Exercise

In this exercise you will browse the toolbars.

1 Display the "tool tip" for each of the buttons on the toolbar in the database window.

2 Select the *Ecostrata* table, then click the *Design* button.

3 Display the "tool tip" for each of the buttons on the toolbar in the design window.

4 Click the *Datasheet* button to display the table in datasheet view.

5 Display the "tool tip" for each of the buttons on the toolbar in the datasheet window.

6 Choose **File|Close** to return to the database window.

Conventions Used in This Manual

Keyboard Conventions

Throughout this manual, whenever instructions are given for the use of keys, they will be enclosed in angle brackets and given the name most commonly used to identify them. Example: <Enter>, <Tab>, <F2>, <Backspace> and <Shift>. When two or more keys are used together, they will be separated with a comma if they must be pressed in sequence and with no comma if they must be pressed in combination. Examples of key usage are indicated below:

<Ctrl><F6>	first key is pressed and held, the second key is tapped
<Ctrl><Shift><F6>	first and second keys are pressed and held, the third key is tapped
<Alt>, <F>, <S>	tap the first key, then tap the second, then tap the third (in sequence)

Typographical Conventions

In this manual, menu commands are in **bold** type and the menu name and option are separated by the (|) symbol (e.g. **File|Save As**). When referring to text that you will see displayed on your screen (except for menus), the text in this manual appears in *italic* type (e.g. *Description* text box).

Procedural Conventions

The word *choose* is used for performing commands from the menu. For example, "choose **File|Save**" means select the File menu (by clicking on the word File or by pressing <Alt><F>) then select the Save option (by clicking on the word Save or by pressing <S>).

When selecting a command button in a dialog box, the word *choose* is also used. For example, "choose *OK*" means that you can click on the *OK* button or (usually) press <Enter>.

The word *select* is used to indicate highlighting text or selecting an option in a dialog box **before** some action is to take place. For example, "in the *Suggestions* list box, select *once*" means that you would highlight the word *once* (probably by clicking on it).

SIA Background

Introduction

Silviculture Information Access (SIA) is a process which takes VM mainframe system data from ISIS and places it into a PC environment for ad-hoc querying and reporting purposes, and for use in other PC based applications such as MS Word, MS Excel and mapping systems. The Ministry of Forests initiated SIA for a number of reasons:

- To reduce system resources used on the VM Mainframe
- To provide districts with the ability to create their own queries and reports, and therefore provide a more flexible reporting environment with more fields available for reporting
- To provide faster reporting times
- To allow for the integration of Silviculture data into other PC applications

What is Information Access?

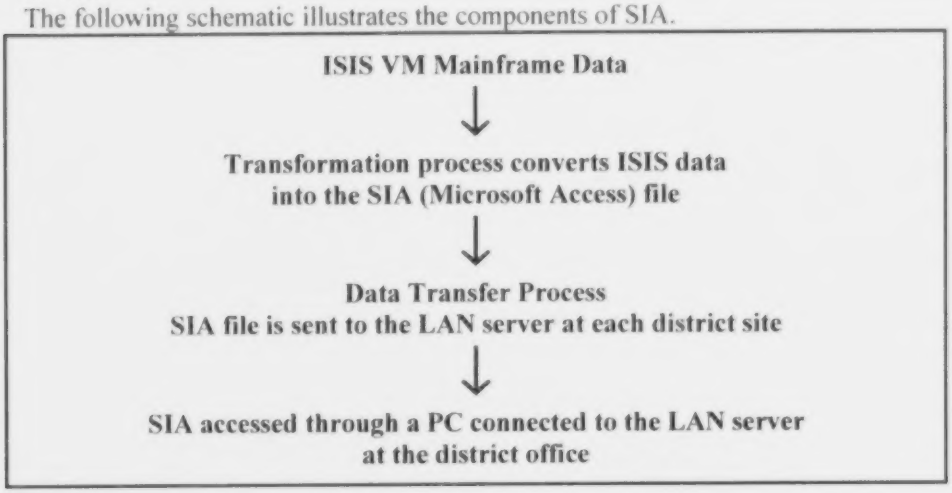
Information Access has two broad components:

Process: - Transformation of information from the VM Mainframe systems into a PC based format

Tool: - Conversion of Information into a PC based file in Microsoft Access format

The Information Access process occurs once every two weeks, and the new data is copied over old data on the LAN server to ensure that the information in SIA is current.

The following schematic illustrates the components of SIA.


ISIS VM Mainframe Data
Transformation process converts ISIS data into the SIA (Microsoft Access) file
Data Transfer Process
SIA file is sent to the LAN server at each district site
SIA accessed through a PC connected to the LAN server at the district office

Differences Between SIA and ISIS Flexible Reports (ISR)

The following table illustrates some of the key differences between the SIA reporting system and the ISR reporting system:

Features and Functions	SIA (PC based)	ISR (mainframe based)
Levels of Reporting	**District**: Excellent **Region**: Poor **Province**: Poor	**District**: Excellent **Region**: Overnight **Province**: Overnight
Report Turnaround Time	**District**: Immediate **Region**: Not Available **Province**: Not Available	**District**: Immediate/Overnight **Region**: Overnight **Province**: Overnight
Training Requirements	**High** Knowledge of Access and Familiarity with data model	**Moderate** Extension of ISIS
Timeliness of Data	Maximum 2 weeks old	Current (1 day old)
Reporting Flexibility	**High** All fields available Unlimited formats	**Low** Fields limited Limited to canned report formats
Data Manipulation Capability	**High** Limited only by MS Access capabilities	**Low** Limited to sorts, and count, min, max, sum, avg
Accessibility	**High** Available on most district PC's	**High** Can log on from any location connected to Office Vision
Building an Interface	**Easy**	**Costly**
Maintenance Requirements	**High** District responsibility for storage of reports/queries	**Low** Maintained by Systems Branch
Saving Reports	**High** Any report/query can be saved for re-use	**Not Available** Report parameters cannot be saved
Data Transfer Capability	**Easy** Cut and Paste available Export to other PC Applications is simple	**Difficult** Many steps are required to translate data into PC format

Relational Database Design

Introduction

The term database is used to describe an organized collection of related information. In a relational database information is stored in a collection of individual *tables* which can be linked together to provide access to all of the information which is stored in the database. In Microsoft Access a database is a collection of tables, forms, queries, reports, macros and modules.

Fundamental to a database table is a common structure which underlines each unit (record) of information. For example, in a telephone book, each record contains a name, an address and a telephone number.

The structure underlying each record actually defines a table. Each column in the table provides storage for one category of information, and each row contains the complete information for one logical unit (record) of information.

The information in a database can be displayed in a variety of different formats. Forms can be used to display all of the information about *one* record. Tables show information from several records. Queries and reports display subsets of information which have been selected from the database.

Database Management System

The term database management system refers to the set of computer programs (software) which is used to store, retrieve, manipulate and report on the information stored in a database.

Database management software allows you to:

- Create the tables to hold the database information
- Create data entry forms for data input and editing
- Sort the data so that it can be displayed in any desired order
- Search for related information using pre-defined criteria
- Link tables together to provide access to all of the information stored in the database
- Establish queries to select subsets of data for display, further processing, and reporting
- Perform a variety of calculations

- Create summary statistics and cross-tabulations to summarize the data

Create report forms on which to print the results of queries and calculations

Database Terminology

Field

General categories of information are called *fields*. The name given to each category of information is referred to as the *field name*.

Each column in a database table is a field. For example, in a telephone directory, the *Last Name* is one field, the *First Name* is another field, the *Address* is a third field and the *Telephone Number* is a fourth field.

Each field in a table has certain *properties* which define the type of information that can be stored in the field as well as the type of operations which can be performed on that data. For example, a field may be defined as character, numeric, date or logical. Numeric fields allow for calculations, date fields are used to record dates, and logical fields restrict the data to a true or false format. Assigning a data type to a field reduces the potential for data entry errors and limits the type of operations which can be carried out on the data. For example, you can perform division calculations only on numeric fields.

Knowing the properties of a field is important for entering selection criteria for the field and for using it to create calculated (virtual) fields.

Record

One logical unit of information (one row in a table) is referred to as a record. Each record (row) contains one complete unit of information, thus one record in the telephone book identifies all of the information about one subscriber to the telephone system. In the Silviculture Information Access database one activity is a record.

Table

A collection of records are stored together in a table. A relational database stores information in several tables which can then be linked together for ease of access. Each table in a relational database contains the information (records) which are specific to one subject area or one purpose. For example, in the Silviculture Information Access database the Ecostrata table stores ecology information only and the Activity Treatment Unit table stores information about ATU's only.

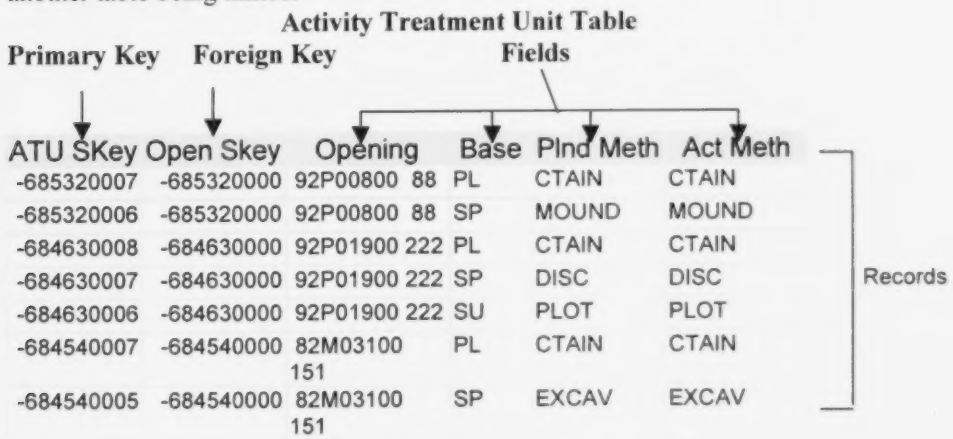
Key

One field in a database table contains information which is unique to each individual record. A typical key field might contain a unique customer number, account number, catalogue number, or Social Insurance Number. The key field is

used to differentiate one record from all others in the table, and can be used to index (order) the records in the table. Key fields are important because they allow you to search for information very quickly, and they also make sure that the integrity of each record is maintained.

Primary and Foreign Keys

Each table in the database can have more than one key field. The *Primary key* is the field which uniquely identifies each record within the table. *Foreign keys* are used to create linkages with other tables. A table can be linked to another table using the foreign key. The primary key in a table will also be the foreign key in another table being linked.



Surrogate Keys

In the SIA database the primary keys are also known as surrogate keys. These key fields contain an unique number which is assigned by the system. Each of the main tables has a primary key which has been system assigned. This key is numeric, and the table is indexed (i.e. ordered) on this field to provide an efficient method of searching a table quickly when it is linked to other tables. When you work with SIA it is important to link tables using surrogate keys because they make the queries much more efficient to run.

Each table in SIA contains a surrogate key which is the primary key for the table, and also contains other surrogate keys from other tables which are used to create the linkages. The primary surrogate key provides an unique identifier for each record in the table, while the other surrogate keys may be duplicated in the table if the link to another table is a one-to-many relationship.

Linking Tables in a Relational Database

Traditionally database information was stored in a *flat file* — a single table which stored all of the information to be tracked by the database. A relational database reduces the complexity of a database and increases both flexibility and efficiency by keeping information in a collection of tables which can then be linked together as needed.

In the Silviculture Information Access database, 26 tables are used to keep subject areas separate from each other. For example, the *Opening table* tracks the opening number, mapsheet, timber mark, cut block and other information related to the opening, the *Activity Treatment Unit table* tracks the treatment unit, the base, technique and method, and other information related to activity treatment units, and the *Ecostrata table* tracks ecology information. These tables can be linked together to combine data from each table into a query to obtain a subset or summary of the information contained within each of these tables. One example of linking tables together is linking an opening to its activities.

Linkages

Tables are linked together (joined) when you need to extract information from more than one table to answer a specific question. Links are the physical joins which are made between tables. These joins allow the records in one table to be correctly matched with the records in another.

Tables can only be joined when there exists a common field which contains the same data in both tables. Tables are joined using the common field, and the records are matched through this common key.

In the SIA database, tables can be linked via the surrogate keys to produce a report. The diagram below shows how the Opening, Stocking Standard and ATU tables can be linked to produce a report which combines some of the data which is stored in these separate tables.

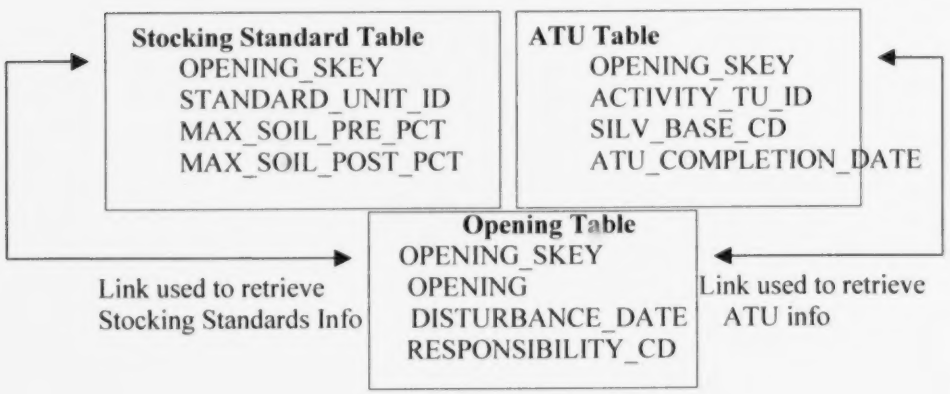


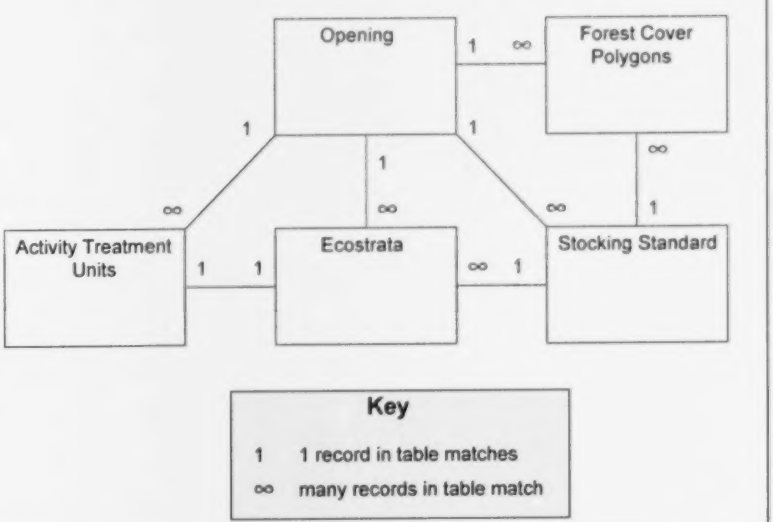
Table Relationship Diagrams

An table is used to store information about a subject area or a concept which you want to keep information about to support the activities of a business.

An relationship diagram is a graphical representation of the relational database model — it shows the tables, and the relationships between the tables (i.e. how they can be joined together).

The diagram below illustrates the relationships between the five main subject area tables in SIA.

Relationships Between the Five Main Subject Area Tables

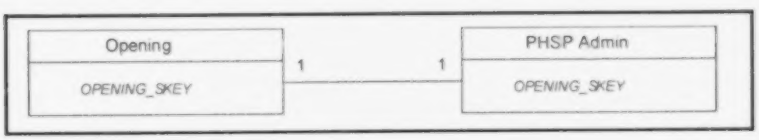


Types of Relationships

When tables are joined together in a relational database, different types of linkages can be established depending on the contents of the tables to be linked.

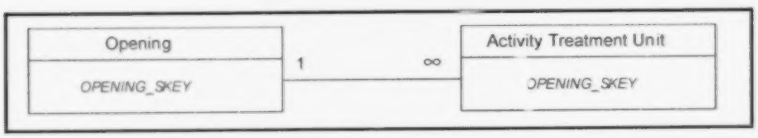
One to One

In a one to one relationship one record in the first table links to just one record in the second table. For example, for each *Opening* there is only one Prescription.



One to Many

In a one to many relationship one record in the first table links to many records in the second table. For example, for each *Opening* there can be one or more *Activity Treatment Units*.



Opening	ATU
93L04000 30	W0
93L04000 30	C00
93L04000 30	E00
93L04000 30	D00
93L04000 30	C01
93L04000 30	C03
93L04000 31	W0
93L04000 31	C01
93L04000 31	E00
93L04000 31	C02
93L04000 31	C03
93L04000 31	C04

Mandatory vs Optional

In a multi-table relational database, some relationships are mandatory, while others are optional. Business rules determine which relationships are mandatory and which are optional.

In the SIA database the link between the *Opening* table and the *Prescription* table is a mandatory relationship. This ensures that all rows in the *Opening* table have associated disturbance type information in the query output. .

The link between the *Opening* table and the *Activity Treatment Unit* table is an optional link because not all Openings have ATU's. This means that rows which do not have an associated ATU will be dropped from query output.

Query

A query is a question you ask about your data. It is a set of instructions and criteria that select just the information you are interested in and present it in a useful format.

For example, you could ask:

- Which openings are not free growing and do not have any activities planned?

Setting Up Your PC

Introduction

Before using the SIA database for the first time, you will need to set up your PC. This will involve creating your own database and attaching the SIA tables to it, and importing the sample queries, forms, reports, macros and modules which form part of the SIA database.

Creating your SIA Database

To work with the SIA database, you must first create your own Microsoft Access database, then attach the SIA tables which are stored on the district LAN server to it. You will also need to import any sample queries, forms, reports, macros and modules which form part of the SIA database.

There are several reasons for setting up this way:

- By creating your own database, you will have a space to store your own queries and reports on your hard drive.
- By attaching the tables stored on the LAN to your personal database, your own database (i.e. queries and report formats) will not be overwritten when the new database is transmitted from Victoria.
- When new data is transmitted from Victoria, you will already be attached to it, so your database will always point to the most recent information.
- The information stored on the LAN cannot be altered.

Exercise - Creating your Database

In the following exercise you will create a database, attach to the tables on the LAN server, and import the sample queries.

1 If necessary, create a directory on your hard disk to store your database.

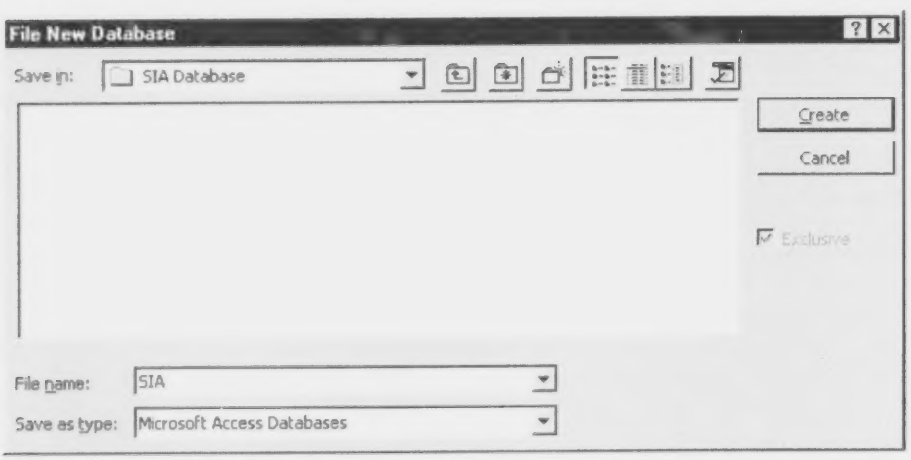
2 Start Microsoft Access.

3 Choose **File | New Database**, then click *OK*

4 In the *File name* textbox, type a name for your new database.

5 Choose the drive that you want to store the database on (usually C: or H:)

6 Select the directory to be used for your database. The *File New Database* dialog box should be similar to the screen capture shown below. Click *create* to create the database.



7 Access will display the *Database* window. Notice that there are no tables currently stored in this database.

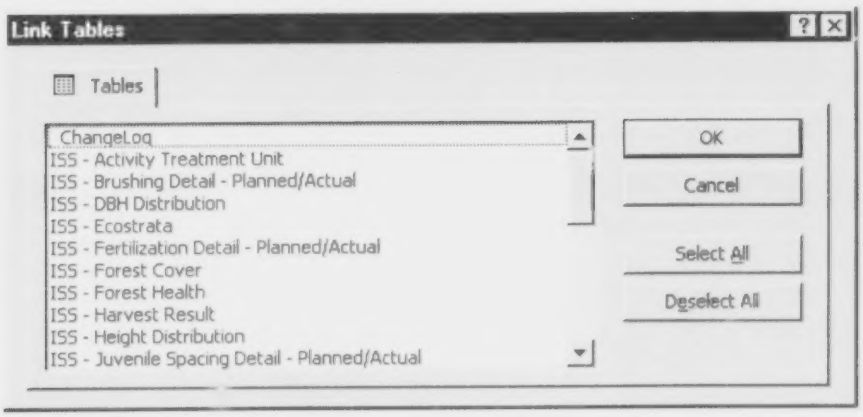
Exercise - Linking Tables

In the following exercise you will link the tables in the SIA database file on the LAN to your personal database.

1 Choose **File | Get External Data | Link Tables**

2 Select the drive and directory containing the SIA database on the LAN server. If you are not sure where the file is stored ask your systems support person. You may also have to be given access to a new network drive. Your file will be called ???SIL80.MDB, where ??? is your organizational unit code such as DCR. Click the *Link* box.

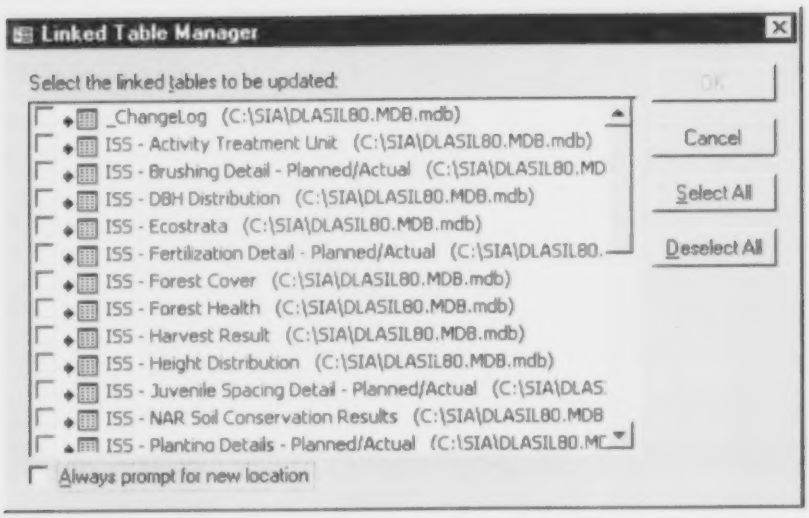
3 Access will present the *Link Tables* dialog box as shown below.



4 Click on the *Select All* box and then *OK* to attach the new SIA tables into your MDB file. There are currently 26 tables (there is also a Change Log where changes to tables, queries etc. are documented)

Verifying and Updating your Linked Tables

5 The *Linked Table Manager* Add-in allows you to check and update your linked tables. To access the Linked Table Manager, choose **Tools | Add-ins | Linked Table Manager**. Access will display the Linked Table Manager dialog box. Use this dialog box to update your linked tables when (and if) necessary. Click *OK* once you have selected the tables to be updated. (Please see image on the next page)


Linked Table Manager
Select the linked tables to be updated:
_ChangeLog (C:\SIA\DLASIL80.MDB.mdb)
ISS - Activity Treatment Unit (C:\SIA\DLASIL80.MDB.mdb)
ISS - Brushing Detail - Planned/Actual (C:\SIA\DLASIL80.MD
ISS - DBH Distribution (C:\SIA\DLASIL80.MDB.mdb)
ISS - Ecostrata (C:\SIA\DLASIL80.MDB.mdb)
ISS - Fertilization Detail - Planned/Actual (C:\SIA\DLASIL80.
ISS - Forest Cover (C:\SIA\DLASIL80.MDB.mdb)
ISS - Forest Health (C:\SIA\DLASIL80.MDB.mdb)
ISS - Harvest Result (C:\SIA\DLASIL80.MDB.mdb)
ISS - Height Distribution (C:\SIA\DLASIL80.MDB.mdb)
ISS - Juvenile Spacing Detail - Planned/Actual (C:\SIA\DLAS
ISS - NAR Soil Conservation Results (C:\SIA\DLASIL80.MDB
ISS - Planting Details - Planned/Actual (C:\SIA\DLASIL80.MC
Always prompt for new location
OK
Cancel
Select All
Deselect All

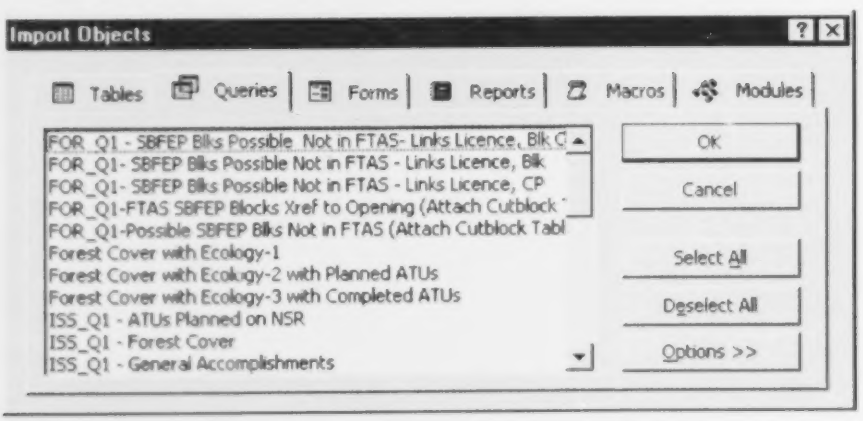
Exercise - Importing Sample Queries, Custom Forms, Reports, Macros and Modules

In the following exercise you will import sample queries, custom modules, Reports, Macros and Forms from the SIA database file on the LAN into your personal database.

1 Choose **File|Get External Data|Import**.

2 Select the drive and directory containing the SIA database on the LAN server. If you are not sure where the file is stored ask your systems support person. You may also have to be given access to a new network drive Your file will be called ???SIL80.MDB, where ??? is your organizational unit code such as DCR.

3 Access will present the *Import Objects* dialog box.


Import Objects
Tables
Queries
Forms
Reports
Macros
Modules
FOR_Q1 - SBFEP Blks Possible, Not in FTAS- Links Licence, Blk C
FOR_Q1- SBFEP Blks Possible Not in FTAS - Links Licence, Blk
FOR_Q1- SBFEP Blks Possible Not in FTAS - Links Licence, CP
FOR_Q1-FTAS SBFEP Blocks Xref to Opening (Attach Cutblock
FOR_Q1-Possible SBFEP Blks Not in FTAS (Attach Cutblock Tabl
Forest Cover with Ecology-1
Forest Cover with Ecology-2 with Planned ATUs
Forest Cover with Ecology-3 with Completed ATUs
ISS_Q1 - ATUs Planned on NSR
ISS_Q1 - Forest Cover
ISS_Q1 - General Accomplishments
OK
Cancel
Select All
Deselect All
Options >>

4 Select the *Queries* tab

5 Select the *Select All* button, then click *OK*

6 Repeat steps 1 through 5 for Forms, Reports, Macros and Modules.

Note: Forest Practices Branch will send a note informing of any changes and updates to Tables, Queries, Forms, Reports, Macro's and Modules. You can also refer to the _Change Log Table – in the SIA Database for updates to these objects.

The SIA Tables

Introduction

The following section lists each table in the SIA database and describes the data stored in the tables, the links to other SIA tables, where the corresponding information is stored in ISIS, and any idiosyncrasies associated with the table.

SIL - Opening Table

The SIL - Opening table contains the information which defines an area of land on which harvesting or a disturbance has occurred. It provides general information about an opening or obligation similar to the information found on the opening inquiry screens in ISIS. Each record in this table contains the information for one opening or obligation.

Some of the information listed below is stored in this table:

- Region and District organization codes
- Mapsheet and opening numbers
- Obligation IDs
- Licensee information
- Opening type and status
- Location Code and Admin Zone
- Gross and Net area for the opening
- Date and type of disturbance
- Previous land type and timber type
- Age, height and stocking class
- Land status and Biogeoclimatic data
- Extract date, update userid and update time stamp

Idiosyncrasies

Ecostrata information is based on the largest ecostrata within the opening.

Land Status information is based on the most recent survey.

Opening Category for Major License openings have an Opening Category of FTML.

Surrogate Keys

The SIL - Opening table can be linked via a surrogate key field to data in other tables. The surrogate key is automatically assigned by, and provides a unique identifier for each opening so that it can be correctly linked to the corresponding records in other SIA tables. This key is duplicated in other SIA tables for the purpose of linking tables together. The name given to this surrogate key is *OPENING_SKEY*.

The corresponding ISIS screens are shown below:

ISIS Screens:

```
        Add       Update    Delete    View      eXit      Help
--------------------------------------------------------------------------------
ISS10M01        Opening Inquiry (OPE)          M View     V 1.00.00  DB SQLP1
 Org Unit..... DKA Kamloops Forest District                      Screen  1 of 13
 Opening..> 92P 008 0.0   88  Licence No.> A28765 CP.>     Blk.> 1
 Timber Mark..> 28765  Blk.> 1                Location. TSINTSUNKO LAKE
 Open Category. FTSBF   1 Status. DFT A     TAUP(ha).   15.3 T NAR(ha).    15.3
 Tenure Type... TSL S20 Competitive    Block Status. Silviculture
--------------------------------------------------------------------------------
 Open Cmt. Y  ExTenure Cmt. N  Multi-Tenure. N  SP Exempt. N
 Client/Locn/Name...> 00027754 00 APPLIED TIMBER MANAGEMENT FTAS(ha).    26.7
 Adm. Forest Org..... DKA Kamloops                          SB Project... 01749
 Management Unit Type U   TIMBER SUPPLY AREA
 Management Unit Id.. 11  Kamloops TSA
 Tenure Admin Org.... DKA Kamloops Forest District
 SP Approval Date....             Actual  Sale Date....... 1991 09
 Disturbance Type.... L           Actual  Harvest Start... 1991 09
 Harvest Offset   = + 0           Actual  Silv Handover... 1993 09
 Responsibility...... F           Prescribe Regen Delay... 1995
 ATU's Scheduled..... Y           Prescribe Free GR: Early 2000  Late. 2004
 SIL150I Small Business Opening information displayed
Command ===>
 F14=SP       F15=MultiTen F16=Area     F17=FTAS     F18=SoilDist F19=LandStat
 F20=RG FG    F21=ForCover F22=StkStds  F23=OthrKeys F24=ForHObs
```

```
       Update    View       eXit       Help
--------------------------------------------------------------------------------
ISS11M01        Opening Inquiry/Part 2 (OPE)  M View     V 1.00.00  DB SQLP1
 Org Unit.....  DKA Kamloops Forest District                    Screen  2 of 13
 Opening..> 92P 008 0.0   88 Licence No.> A28765 CP.>     Blk.> 1
 Timber Mark... 28765   Blk.. 1            Location. TSINTSUNKO LAKE
 Open Category. FTSBF   1 Status. DFT A    TAUP(ha).   15.3 T NAR(ha).    15.3
--------------------------------------------------------------------------------
 Previous Stand Type... MAT  Timber Type..... PLI        6 3 3
 Timber Supply Area/Blk 11B Kamloops TSA Block 2
 Seed Planning Zone.... TOD Thompson Okanagan Dr  Community Watershed... N
 Photo/Line No(s)..... BC86091-54
 Field Work By........ F.C.I.
 UTM:            Zone.. 10      East... 673300     North.. 5659300  NAD. 27
 Lat/Long:   Latitude..         Longitude..
 Treatment Regime Id..> D   CLEARCUT/PLANT      Admin Zone.> E   ZONE E
 Location: Code 1.> 49 TSINTSUNKO LAKE       Code 2.>
 Funding Source....... SBF Small Business Forest Enterprise Fund
 Last Updated By...... LAWOOD   WOOD, LORNE A             Date... 1997 10 06
 Opening Comments..... Y  Cross Reference.                  SX Trial Cmnt. N
 SIL150I Small Business Opening information displayed
Command ===>
 F14=SP       F15=RecAct   F16=PrjMgmt  F17=Audit    F18=SoilDist F19=LandStat
 F21=ForCover F22=StkStds  F23=OthrKeys F24=Sendnote
```

Exercise — Comparing the SIL - Opening Table to ISIS

1 Make sure that your personal *MDB* database window is open.

2 Double-click the *SIL - Opening* Table to display the records in Datasheet view. Notice that the data is displayed in a tabular format compared to the ISIS screen shown above which show the data in a Form layout.

3 Locate the *Surrogate key* field. Notice that each record has a unique number in this field to identify the opening. The surrogate key numbers are in descending order. The actual *Opening* field consists of a combination of the Mapsheet and the Opening number, however these are not in a particular order. The surrogate key provides a faster method of locating a particular opening when tables are linked together.

4 Switch to design view.

5 Locate the fields in the *SIL - Opening* table which match the fields in the ISIS screen shown above.

6 Close the *SIL - Opening* table to return to the database window.

ISS - Activity Treatment Unit Table

The ISS - Activity Treatment Unit (ATU) table contains the information which defines ISIS activity treatment units within an opening. It provides a detailed listing of all Silviculture treatment units, both planned and completed. Each record within this table contains one planned or completed ATU.

Information stored in this table includes:

- Region and District organization codes
- Opening Identifier
- Some opening level information
- Base, technique and method
- Silviculture objectives
- Planned and completion dates for the ATU
- Planned and completed area for the ATU
- Planned and completed cost for the ATU
- Planned and completed funding source for the ATU
- Ecostrata information

Idiosyncrasies

One particularly important field in the ATU table shows whether or not activity results have been recorded for each Activity Treatment Unit. This information is stored in the ***RESULTS_IND*** field in the form of a "Y" if the ATU is finished, and an "N" if it is planned. This is similar to blank or 'R' on the ATU schedule screen.

The Activity Treatment Unit table is very large. If you are only interested in one type of activity (i.e. planting, stand tending, etc) do not use this table. Use one of the detail tables instead.**Note:** Since both planned and actual data is stored in this table ensure you select the correct fields. Planned data items start with PLND or PLANNED and completed data items start with ACT or ACTUAL.

Surrogate Keys

The ATU table is linked via surrogate key fields to data in other tables. The primary key for this table is the *ACTIVITY_TU_SKEY* field. This key provides a unique identifier for each treatment unit so that it can be correctly linked to the corresponding records in other SIA tables.

The ATU table contains two other surrogate keys. These keys are foreign keys in the ATU table but are the primary keys in the other tables which can be linked to it. These keys are used to link the ATU table to the SIL - Opening table, and the ISS - Silv Project table.

The ISIS screen which corresponds to the ATU table is shown below:

```
        Update     View       eXit       Help
--------------------------------------------------------------------------------
PAG01M01          ATU Schedule (SCH)           M View      V 1.00.00  DB SQLP1
 Org Unit...... DKA Kamloops Forest District
 Opening..>  92P 008 0.0    88  Licence No.> A28765 CP.>      Blk.> 1
 Timber Mark... 28765     Location. TSINTSUNKO LAKE            FTAS(ha).     26.7
 Open Category. FTSBF    1 Status. DFT A     TAUP(ha).    15.3 T NAR(ha).     15.3
                              App Date             Actual  Sale Date..... 1991 09
 Declared Regen Delay. 1992                        Actual  Harvest Start. 1991 09
 Prescribe Free GR: Early> 2000 Late.> 2004        Actual  Silv Handover. 1993 09
--------------------------------------------------------------------------------
Comments.. N    *-- Activity> --* SX Trial Cmnt. N          ATU   1 to   6 of   6
Opt Rslt  ATU  Base Tech  Method Obj   Date  Cmt P R   Area  Fund OPYr   Proj
      R    X0   SP   ME   MOUND  PL   1993 09 Y  M M    15.3  SBF 1993 SP1993
      R   S01   SU   PH          PH   1994 04 N  H H    15.3  SBF 1994 SU1995
      R    Y0   PL   PL   CTAIN  CE   1994 07 Y  M M    15.3  SBF 1994 PL1995
      R   S02   SU   RG   PLOT   RG   1997 09 N  H H    15.3  SBF 1996 SU1998
          S04   BR   MA   POWER  CGE  1999 07 N  H H    15.3  SBF 1999
          S03   SU   FG   PLOT   FG   2002 07 N  H H    15.3  SBF 2000

 MSG004I You have reached the bottom of the list
Command ===>
 F11=Reports  F12=Cancel   F14=SP       F15=RecAct   F16=PrjMgmt  F17=Audit
 F18=Details  F19=LandStat F20=TreatReg F21=ForCover F22=SortATU  F23=OthrKeys
```

Exercise — Comparing the ATU Table to ISIS

1 Make sure that your personal *MDB* database window is open.

2 Double-click the *ISS - Activity Treatment Unit* table to display the records in Datasheet view. Each record has a unique number in the *ACTIVITY_TU_SKEY* field to uniquely identify each treatment unit. The *ACTIVITY_TU_SKEY* numbers are in descending order.

3 Notice the surrogate keys in this table. The numbers in these foreign keys are not unique within this table. An Opening may have several Activity Treatment Units. Not all ATU's have Silviculture projects attached to them, so the *PROJ_SKEY* may contain a zero value.

4 Switch to design view.

5 Locate the fields in the ATU Table which match the fields in the ISIS screen shown above.

6 Close the ATU Table to return to the database window.

ISS - Forest Cover Table

The ISS - Forest Cover table contains the information which defines the forest cover within a specific polygon based on the most recent survey. Each forest cover polygon and polygon layer (i.e. Silviculture or Inventory) is entered as a separate record.

Information stored in this table includes:

- Region and district organization codes
- Opening Identifier
- Some opening level information
- Polygon number, layer and rank
- Tree species within each polygon
- Stocking type, status and age
- Total and well spaced stems per hectare

Idiosyncrasies

One important field in this table is the *FOR_COVER_LAYER_CD* (Forest cover layer code) which identifies whether the layer is an inventory layer or a Silviculture layer. When building queries, you must be aware of the separate layers. Generally, you should specify one layer or the other to prevent double-counting on the polygon. Also be aware of multi-layer stands (i.e. 1S, 2S, 3S, 4S). Major licensees do not specify a polygon number. They submit a strata ID which is stored in the polygon number field of ISIS.

Surrogate Keys

The ISS - Forest Cover table is linked via surrogate key fields to data in the other tables. The primary key for this table is the *FOREST_COVER_SKEY* field. This key provides an unique identifier for each forest cover definition within a specific polygon so that it can be correctly linked to the corresponding records in other SIA tables.

The ISS - Forest Cover table contains two other surrogate keys. These keys are used to link the ISS - Forest Cover Table to the SIL - Opening table and to the ISS - Stocking Standards table.

The ISIS screens which corresponds to the ISS - Forest Cover table are shown below:

```
SUR62M01      Opening Forest Cover Summary(FCO) V1.00.00 DBSQLP1
Org Unit....DKAKamloops Forest District
Opening..> 92P008   0.0          88          Licence No.>        A28765   CP.>                Blk.>      1
Timber Mark.>        28765        Blk.>      1
--------------------------------------------------------------------------------
Last Action Date.1997 10 06   Comments.. N                Lines  1        to        3        of        3
   Poly S                                 Stocking        Ref  *-------Species-------* *-----------Stems/ha----------*      F
 SU   Strt   F   Lyr   Rnk   Area  Stat    Type  Year  Code   Code   Code   Dens   TWS     WS      FG      G
      644   A                  .3  NP      RD    1994                                                    N
 1    644                     15.3  IMM     ART   1997  BL     PLI            9547                       N
 1    644        S            15.3  IMM     ART   1997  PLI                                    1147       N


MSG0041 You have reached the bottom of the list
Command ===>
F11=Reports     F12=Cancel     F13=ATU        F14=SP         F15=RecAct          F16=PrjMgmt
F17=Audit       F18=Detail     F19=LandStat   F21=ForCover   F23=OthrKeys        F24=Sendnote
```

```
          Update              View              eXit              Help
--------------------------------------------------------------------------------
SUR60M01      Opening Forest Cover      (OFC)     M  View          V  1.00.00     DB  SQLP1

Opening..>      92P      008     0.0     88       Licence No.>A28765          CP.>       Blk.>      1
Location.....   TSINTSUNKO LAKE                   TAUP(ha).                   15.3  T    NAR(ha).   15.3
--------------------------------------------------------------------------------
Last Action Date..    1997 10 06                  Comments..      N                   Lines  3      of        3
 Option(A,C,U,D)..
 SU>. 1  Polygon/Strata    644   Sfx.              Poly Xref.                                Layer.  S      Rank.
 Area (ha).....          15.3      Stocking Status.    IMM       Stocking Type.    ART
 Species Code/%.       PLI       100       %                 %                   %                        %
 Age (yrs):  Average.  3           Planted.  1                 Range.
 Height (m).....       .4          Crown Closure (%).          %  Ref Year (YYYY).  1997
 Site Index.....       16          Site Index Source.          Site Class...     M
 Stems/ha:  Density.               Total WS.         WS.  1147 FG..
 Reserve Type...                   Basal Area (m2/ha).         DBH.  1
 VAF.....                          Data Source.  11    Origin.  Stocking Class.                    FG       N
 AdjMap.....           N

MSG0041 You have reached the bottom of the list
Command ===>
F11=Reports     F12=Cancel     F13=ATU          F14=SP          F15=RecAct        F16=PrjMgmt
F17=Audit       F18=AdjMap     F19=LandStat     F23=OthrKeys    F24=ForHObs
```

Exercise — Comparing the ISS - Forest Cover Table to ISIS

1 Make sure that your personal *MDB* database window is open.

2 Double-click the *ISS - Forest Cover table* to display the records in Datasheet view. Each record has an unique number in the *FOREST_COVER_SKEY* field to uniquely identify each forest cover definition. The *FOREST_COVER_SKEY* numbers are in descending order.

3 Notice the two other surrogate keys in this table. The numbers in these foreign keys are not unique. An Opening may have several Forest Cover definitions, and several forest cover definitions may have the same stocking standards.

4 Switch to design view.

5 Locate the fields in the *ISS - Forest Cover* table which match the fields in the ISIS screen shown above.

7 Close the *ISS - Forest Cover* table to return to the database window.

ISS - Stocking Standard Table

The ISS - Stocking Standard table contains the information which defines the stocking standards for an area of land on which harvesting or a disturbance has occurred. The Stocking Standard table provides information on the stocking standard for each standards unit by opening. Each different standard unit is a separate record in this table.

Information stored in this table includes:

- Region and District organization codes
- Opening Identifier
- Some opening level information
- Regen method to be used
- Net area to be reforested
- Regen delay
- Free growing indicators

Surrogate Keys

The ISS - Stocking Standard table is linked via surrogate key fields to data in the other tables. The primary key for this table is the *STOCKING_SKEY* field. This key provides an unique identifier for each stocking standard so that it can be correctly linked to the corresponding records in other SIA tables. The *STOCKING_SKEY* is used to link this table to the ISS - STOCKING LAYER table which has information about 1 or more layers associated with the stocking unit.

The ISS - Stocking Standard table contains one other surrogate key. This key is used to link the ISS - Stocking Standard Table to the SIL - Opening Table.

The ISIS screen which corresponds to the ISS - Stocking Standard table is shown below:

```
        Update     Delete     View       eXit       Help
--------------------------------------------------------------------------------
PHP46M01              Prescription (NEW)       M View      V 2.00.00  DB SQLP1
Stocking Requirements                                          Screen 9  of 13
Opening..>  92P 008 0.0    88  Licence No.> A28765 CP.>      Blk.> 1
Open Category. FTSBF    1 Status. DFT A     TAUP(ha).    15.3 T NAR(ha).     15.3
--------------------------------------------------------------------------------
                                                                SU   1 of   1
SU..................... 1     Net Area to be Reforested..    15.3
Regen Method.......... PLANT Plant
Regen Date(yrs)......  4      Free Growing(yrs):  Early.  9 Late. 13

Max Coniferous(st/ha)....
Post Spacing Density(st/ha):        Min.              Max.
PreferredSpecies/Height(m).. SX         PLI
AcceptableSpecies/Height(m).
Well Spaced Trees/ha: Target. 1200 Min.  700 Min Pref.       Min Horiz(m).
Planned Residual Basal Area(m2/ha) Min.             Max.        Prune Hght.
Height Relative to Comp(cm/%). 120 %        Other Requirements Comments... N
Special Survey Criteria Comments.. N        Rationale for Choice Comments. N
 SIL150I Small Business Opening information displayed
Command ===>
 F13=ATU       F14=PrscSum  F15=Tenure    F16=Area      F17=MgtObj    F18=Site
 F19=Soil      F20=SilvSys  F21=Refor     F22=Layers    F23=OthrKeys F24=Sendnote
```

Exercise — Comparing the ISS - Stocking Standard Table to ISIS

1 Make sure that your personal *MDB* database window is open.

2 Double-click the *ISS - Stocking Standard table* to display the records in Datasheet view. Each record has a unique number in the *STOCKING_SKEY* field to uniquely identify each stocking standard. The *STOCKING_SKEY* numbers are in descending order.

3 Notice the other surrogate key in this table. An Opening may have several stocking standards.

4 Switch to design view.

5 Locate the fields in the *ISS - Stocking Standards* table which match the fields in the ISIS screen shown above.

6 Close the *ISS - Stocking Standards* table to return to the database window.

ISS - Ecostrata Table

The ISS - Ecostrata table contains the biological information for an area of land on which harvesting or a disturbance has occurred. Each eco stratum is stored as a separate record. This table provides information for one or more eco stratum for each standard unit.

Information stored in this table includes:

- Region and District organization codes
- Opening Identifier
- Some opening level information
- Gross and net area
- Biogeoclimatic data
- Site series and type
- Elevation, aspect and slope data
- Soil characteristics

Idiosyncrasies

For older openings the standards unit is system generated.

Surrogate Keys

The ISS - Ecostrata table is linked via surrogate key fields to data in the other tables. The primary key for this table is the *ECO_STRATA_SKEY* field. This key provides a unique identifier for each eco stratum so that it can be correctly linked to the corresponding records in other SIA tables.

The ISS - Ecostrata table contains two other surrogate keys. These keys are used to link the ISS - Ecostrata table to the SIL - Opening table and to the ISS - Stocking Standard table.

The ISIS screen which corresponds to the ISS - Ecostrata table is shown below:

```
        Update    Delete    View      eXit      Help
-------------------------------------------------------------------------------
PHP33M01             Prescription (NEW)      M View     V 2.00.00  DB SQLP1
Ecology and Critical Site Factors                                 Screen  4 of 13
Opening..>  92P 008 0.0    88  Licence No.> A28765 CP.>      Blk.> 1
Open Category. FTSBF     1 Status. DFT A    TAUP(ha).    15.3 T NAR(ha).    15.3
-------------------------------------------------------------------------------
Previous:Type. MAT  Species. PLI        Age. 6  Hght. 3  Stock. 3  Site.
Option(A,C,U,D).....                                          Stratum   1 of   1
SU.................> 1      ECO Stratum.  A  Moisture.      Nutrient.
BGC Zone/Sub/Var&Ph.> MS   dm  2   Site Series/Type....> 03
Elevation(m):     Min.         Max.         Avg. 1600
Slope:         Aspect. V  Variabl Min%.      Max%.      Avg%.  10  Pos.
Soil:        Parent Material.      Surface.         Subsurface.
 CF%: Min.      Max.        Restricting Type.
 Unfav. Substrate: Min.       Max.        Restrict Layer: Min.        Max.
Hydrology: Min Depth to HW.      Max Depth to HW.      Seepage (Y/N). N
Humus:           Form.        Min Depth.      Max Depth.       Land Form.
Critical Site Factors.. N
Ecology Comments..... Y
 MSG022I No data retrieved from search
Command ===>
 F13=ATU      F14=PrscSum  F15=Tenure   F16=Area     F17=MgtObj   F18=Site
 F19=Soil     F20=SilvSys  F21=Refor    F22=StkStds  F23=OthrKeys F24=Sendnote
```

Exercise — Comparing the ISS - Ecostrata Table to ISIS

1 Make sure that your personal *MDB* database window is open.

2 Double-click the *ISS - Ecostrata* table to display the records in Datasheet view. Each record has a unique number in the *ECO_STRATA_SKEY* field to uniquely identify each ecostrata. The *ECO_STRATA_SKEY* numbers are in descending order.

3 Notice the two other surrogate keys in this table. An Opening may have several stocking standards which may have several ecostrata.

4 Switch to design view.

5 Locate the fields in the *ISS - Ecostrata* table which match the fields in the ISIS screen shown above.

6 Close the *ISS - Ecostrata* table to return to the database window.

Other Tables

SIA contains other tables which can be used to build queries. A quick overview of some of these other tables follows:

ISS – Brushing Detail - (Planned/Actual)

This table provides detailed information on planned and actual brushing activities. This table can be used with the DBH, Height Distribution, and Vegetation Competition tables.

ISS - DBH Distribution

This table is used in combination with the Stand Tending Details Planned/Actual table and the ATU table to provide DBH (diameter at breast height) information for ATU's with stand tending activities.

ISS – Fertilization Detail - (Planned/Actual)

This table provides detailed information on planned and actual fertilization activities.

ISS - Forest Health

This table provides information regarding observed pest occurrences.

ISS - Height Distribution

This table is used in combination with the Stand Tending Details Planned/Actual table and the ATU table to provide height information for ATU's with stand tending activities.

ISS – Juvenile Spacing Detail - (Planned/Actual)

This table provides detailed information on planned and actual spacing activities. This table can be used with the DBH, Height Distribution, and Vegetation Competition tables.

ISS – NAR Soil Conservation Results

This table provides information recorded for observed soil disturbance including road, landing and other degradation, observation date, and the activity which noted the disturbance.

ISS - Prescription

This table provides detailed prescription administration information such as approval and amendment dates, and the names of people who initiated or updated amendments, etc.

ISS - Prescription Forest Health

This table provides a record of the type of pest occurrence observed on a standards unit at the time of the Prescription.

ISS - Prescription Harvest

This table provides harvesting details for an opening including the type of harvesting method(s) used.

ISS – Harvest Detail – Planned/Actual

This table provides detailed information on planned and actual harvesting activities.

ISS - Planting Details (Planned/Actual)

This table provides information on the planned or actual planting activities for an ATU.

ISS - Planting Stock Allotment

This table shows the total number of trees planted (or to be planted) with details about the stock type, species and seedlings for planting ATU's.

ISS - Range Prescription

This table provides Prescription range information for openings.

ISS – Prescription Silvicultural Systems

This table provides the Silvicultural Systems Information for the Prescription.

ISS - Seedling Requirements

This table provides a summary of all the stocking type requests planned for each standards unit. It allows you to relate specific requests to specific openings and standards units.

ISS - Silv Project

This table provides information on a Silviculture project such as the bid price, contract award method, and project co-ordinator.

ISS - Site Preparation Details (Planned/Actual)

This table provides planned and actual site preparation details for an ATU.

ISS – Pruning Detail (Planned/Actual)

This table provides detailed information on planned and actual pruning activities.

ISS - Stocking Layer

This table provides the stocking layer information for each stocking standard unit.

ISS - Vegetation Competition

This table provides information on the vegetation competition for an ATU which had a pre-stand tending survey. It is used primarily for planning stand tending activities and is normally combined with the Stand Tending Details table.

Exercise — Viewing Other SIA Tables

1 Make sure that your personal *MDB* database window is open.

2 Open each of the tables discussed in the section above and study the contents in both datasheet and design view. Knowledge of the tables in SIA is crucial to the development of Queries.

3 Close all open tables and return to the database window.

Sample Queries

Introduction

Looking through the tables in a database is one method of viewing data, however a database can contain massive amounts of data and it can be difficult to see trends or to isolate a specific piece of information using this method. It is also impossible to summarize the information, and to limit the fields and records which are included in the output.

In most cases you will be interested in extracting a smaller subset of data from the database to answer specific questions. Queries provide the mechanism for extracting subsets of data which meet specific criteria that you specify.

A query is a question you ask about your data. It is a set of instructions and criteria that select just the information you are interested in and present it in a useful format.

For example, you could ask:

- Which openings have had root rot disease observed on them?

MS Access gathers the data that answers your question. The data that makes up the answer is called a dynaset or a result set. Each time you run the query, the information in the set is updated.

Query Elements

A query allows you to create a method of filtering and sorting information in the database. Queries include only the data items and rows you select and can also include summary statistics.

Queries allow you to link several tables together to extract information from more than one source, and you can store the query instructions in a file so that the query can be re-used anytime the same output format is required.

Dynasets or Result Sets

Queries generate what are called "dynasets". A dynaset is a dynamic set of records which *looks* like a database table, however it can contain data which has been extracted from several tables. The query contains the instructions (criteria) necessary to create the dynaset, but the records included in the dynaset are not actually stored in a table - instead the query program recreates a "*virtual table*" whenever the query is run. When new data is imported into the database, the query will still operate, but will produce output which is based on the current information stored in the database tables.

Query Types

Several different types of queries can be created to suit a variety of different purposes. The basic types of query are Select Queries and Summary Queries . Specialized query types (Action Queries) are covered in detail later in this guide. The following table details the basic query types and how they work.

Query Type	**Uses**
Select Query (Detail)	Selects fields and records based on the selection criteria used and presents them in a dynaset without changing the data.
Select Query (Summary)	Summarizes or collapses the detailed information into a summary of the data in the dynaset.
Crosstab Query	Creates a summary of data based on two fields in the database in a spreadsheet type format using columns and rows. Provides a more compact format than a select query.
Parameter Query	A parameter query extends the functionality of other query types by allowing you to specify one or more criteria interactively when the query is run.

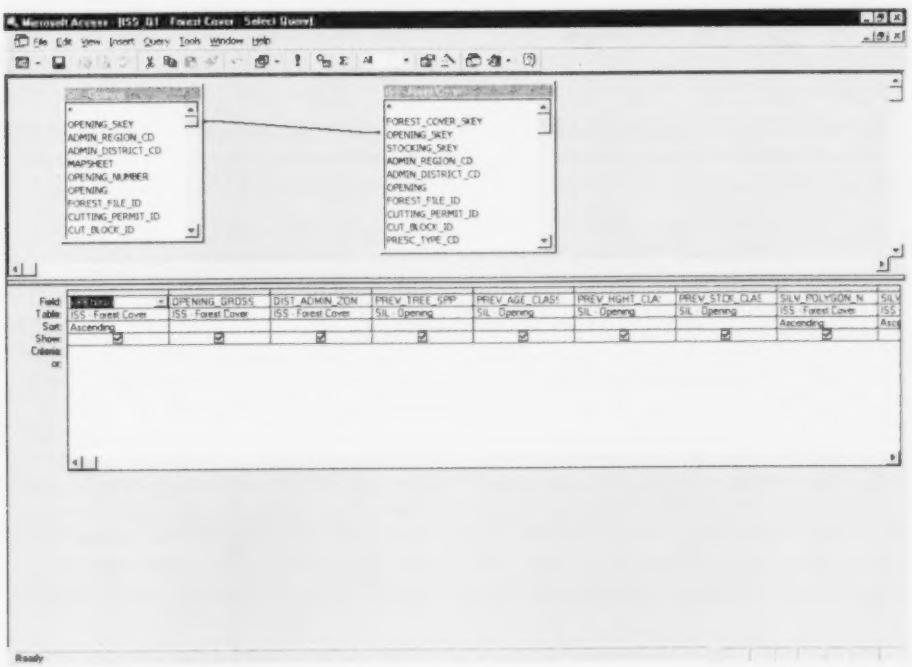
The Query Design Screen

The query design screen consists of two panels.

The upper panel shows all of the tables which supply data to the query output, and also any linkages which exist between these tables.

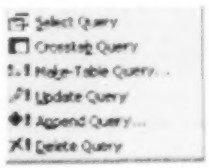
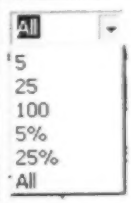
The bottom panel consists of the query grid, which lists the fields to be included in the query output, as well as any sort and record selection criteria.

The query design window also has a toolbar which is specific to tasks which are carried out from this screen. The screen capture below shows the components of the Query Design screen.



The Query Design Toolbar

The query design window contains a toolbar with push buttons for common query design tasks. The table below identifies the purpose of each of the buttons on this toolbar.

Button	Purpose
	These buttons are used to switch between Design view, Datasheet view and SQL view respectively (SQL View displays the SQL program created by the query)
	This button saves the query
	The Query Type Button has a drop down menu which allows you to specify either a Select or Crosstab query and Make-Table, Update, Append, or Delete query
	This button runs the query and displays the query result in datasheet view
	This button allows you to add additional tables to your query.
	This button adds a Totals row to the query grid when you want to add summary statistics to your query.
	The Top Values box, finds the top values in the active query based on a percentage of rows or a number of rows. To find the top values, choose a percentage (such as 5%) or enter a number (such as 25).
	This button displays the Properties dialog box for the selected object in the query window
	This button displays the Expression Builder dialog box
	This button switches you back to the database window

Button	Purpose
	These buttons create a new query, new form or new report based on the current table.

Using the Sample Queries

The SIA database contains some pre-defined queries which you can run against the database or modify by adding new fields and/or new criteria.

These queries can be listed by clicking the *Query* button in the database window. Once the list of queries has been displayed you can open a query in datasheet view (to view the dynaset of records) or in design view (to view and/or modify the query grid).

One-Table Queries

The structure of the SIA database allows you to base many queries on a single table. This provides several advantages. It reduces the complexity of the query, limits the potential for errors and for inaccurate results, and makes the query very efficient to run. When designing your own queries, limit the query to one table if possible, or include only the tables which are absolutely necessary to achieve the desired output.

Exercise - Viewing One Table Sample Queries

1 Display and maximize the SIA database window.

2 Click the *Query* button to display a list of all of the sample queries. Notice that each query name begins with "ISS_Q1 - ". This helps to differentiate between tables and queries.

3 Select the query entitled *ISS_Q1 – ISIS Opening Information.*

4 Choose *Design* to display the query design screen. Notice that this screen is split into two panels. The upper panel shows which tables are included in the query. The lower panel contains the *query grid* which is used to specify which fields are included in the output as well as the criteria which are used to sort and select the output records. Notice that this query is based on a single table.

5 Use the scroll bar underneath the query grid to see which fields are included in the output from this query.

6 Select queries are stored as an SQL program (Structured Query Language). View the contents of the SQL Select statement by clicking the *SQL* button on the toolbar.

7 Click the Query *Design View* button to return to the query grid.

8 Click the *Datasheet* button (or the *Run* button) to run the query. Examine the contents of the output. Notice that only the fields included in the query grid are included in the output.

9 Press <Ctrl><F4> to return to the database window.

10 Examine the Query design, the criteria, and the Query output for the following one-table query:
ISS_Q1 - General Planning

Modifying the Sample One-Table Queries

The sample queries can be modified and resaved, or saved under a new name as a new query. This method can save you time if the sample query has most of the information that you need.

To modify a query, you simply add or rearrange the fields, and add any additional record selection or sort criteria.

Sorting Records

The most basic type of queries are used to select output fields and display the records in the resulting dynaset in sorted order.

When you want to sort the output you simply specify the field which will determine the order of the records in the output.

If the sort field is not the primary key for the table, a sort will often result in records being grouped together in the output. For example a sort by *FOREST_FILE_ID* will result in the records with the same *FOREST_FILE_ID* being displayed together in a group.

To subsort within a group of records additional fields must be used to resolve any ties. You can add as many fields as needed to get the records into the exact order required, however, it is important to note that Access uses a hierarchical sort, meaning that the sort fields are used starting from the leftmost sort field in the query grid, then in order from left to right.

Creating Sort Criteria

a) Open the query in design view.

b) Locate the field to be used for your sort in the query grid.

c) To sort records using the selected field, select *Ascending* or *Descending* in the *Sort* cell in the grid.

d) To perform a multi-level sort specify *Ascending* or *Descending* in the *Sort* cell of the tie-breaker fields in the query grid. Access will sort in hierarchical order reading the columns in the grid from left to right, therefore it is important to have the fields arranged in the grid in the correct order.

e) To apply the sort, click the *Run* or the *Datasheet* button on the toolbar (or choose **Query|Run**).

Exercise - Modifying Sample Queries - Sorting

In this exercise you will sort an existing query to display the records in a variety of different orders, and you will save the sorted query under a new name.

Sorting Records Using a Single Field

1 Open the query entitled *ISS_Q1 – ISIS Opening Information* in design view.

2 Click in the *SORT* cell in the *OPENING* field, and choose *Ascending*. Your grid should look similar to the grid shown below:

Field:	OPENING	FOREST_FILE_ID	CUTTING_PERMIT	CUT_BLOCK_ID
Table:	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening
Sort:	Ascending			
Show:	☑	☑	☑	☑
Criteria:				
or:				

3 Run the query. Notice that the records are now displayed in order by Opening number.

4 Click the design button to return to the query grid, then choose *Not Sorted* in the *Sort* cell in the *OPENING* field.

5 Click in the *Sort* cell in the *DISTURBANCE_DATE* field, and choose *Descending*.

6 Run the query. Notice that the records are now sorted in descending order by disturbance date.

7 Click the design button to return to the query grid, then choose *Not Sorted* in the *Sort* cell in the *DISTURBANCE_DATE* field.

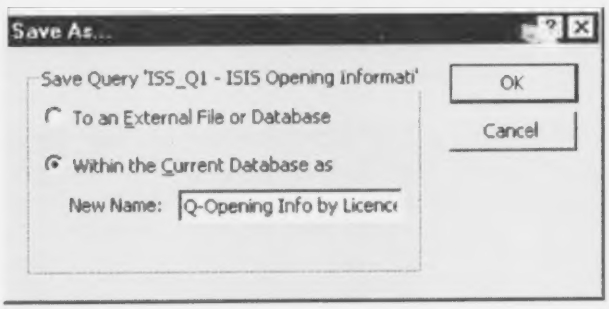
8 Create an ascending order sort by *FOREST_FILE_ID* and run the query. Notice that several records have the same licence number which results in the records being grouped by licence numbers.

9 Return to the query grid (design view).

Multi-level Sort

10 Add a descending order sort to the *OPENING_GROSS_AREA* field and rerun the query. The *FOREST_FILE_ID* is the leftmost field in the grid containing sort criteria, therefore the records are still grouped by Licence number, however, within each group of licences, the openings are sorted in descending order of gross area.

11 Choose **File | Save As/Export...**, and save the revised query as *Q - Opening Info by Licence*. Close the query and return to the database window.



12 Open the query named *ISS_Q1 - General Planning* in design view.

13 Sort it in ascending order using the *SILV_BASE_CODE* and *PLND_TECHNIQUE_CD* fields as the sort keys. Run the query.

14 Return to the query grid, then set the *PLND_DATE* field to a descending order sort.

15 Perform the sort. Which is the primary sort key? _____
What order are the sort keys used? _____
_____

16 Return to the query design grid and choose **File | Save As/Export...**.

17 Type the following name for the query, then click *OK*.
Q - ATUS Sorted by Date/Base/Tech

18 Close the query and return to the database window.

Selecting Records

Record selection criteria must be used when you want to select a specific subset of records for the query output.

Record selection criteria specify the condition (or conditions) under which a record will be included in the query output. Each condition must include a field plus an example of the data in that field. This tells Access *where* to look and *what* to look for. If the data in the field and the example data match then the record is selected for output. It is therefore essential to be very familiar with the fields in the table and the type of data stored in those fields in order to create successful record selection criteria.

Relational (Comparison) Operators

Comparison Operators are used to compare the contents of the specified field with the example of the data that you provide. If the comparison is true the record will be included in the output.

When entering your example data, use the following rules:

- Text values should be entered inside quotation marks and case matched with the data in the field.
- Numeric values can be entered as is with no special treatment.
- Date values should be enclosed in between pound signs (#). Note that the format of example dates should match the way your system so the example dates should be entered in the form YYYY/MM/DD
- Logical Fields are entered as Y or N

The following table lists the comparison operators, an example of how they are entered, and the resulting record selection.

Operator	Meaning	Example	Selected if value is...
=	Equal to	="PL"	Equal to "PL"
<	Less than	<50	Less than 50
<=	Less than or equal to	<=#1991/01/01#	Dated on or before Jan 1, 1991
>	Greater than	>#1999/12/31#	Dated after Dec 31, 1999
>=	Greater than or equal to	>=1500	More than or equal to 1500
<>	Not equal to	<>"PL"	Not equal to "PL"

Other Comparison Operators

There are some additional comparison operators which you can use to compare field values with example values. These operators provide more sophisticated matching criteria to extract the desired records.

Some of these additional operators are listed in the table below:

Operator	Meaning	Example	Selected if value is...
Between	Between two values	Between 15 and 20	In the range 15-20
In	In a set of values	In ("PL", "FP", "RP")	Is PL, FL or RP
Not	Not equal to	Not "Y*"	Does not begin with Y (can also be used with other operators)

Operator	Meaning	Example	Selected if value is...
Like	Like *pattern*. The pattern can include wildcards such as * and ?.	Like " 82F*" (enter a space before the 82)	Picks mapsheets starting with 82F
Is Null	Field is blank	Is Null	Is blank
And	The conditions on either side of And must both be met		
Or	One of the two conditions must be met		

Note: *Mapsheet number in ISIS is designed for 3 character numbers (e.g. 101). For districts who only have 2 character number, you will have to enter a blank/space-leading when specifying mapsheet number in the critiera (e.g. _ 82F03000). ATU Id is also set up as a 3 character field. When fewer than 3 character is required in the criteria, enter blank/space-leading (e.g. _ W1). _ represents a blank space.*

Creating Record Selection Criteria

To create Record Selection Criteria:

a) Open the query in design view. The table fields are listed in the top part of the window, and the criteria grid is displayed below. Record selection criteria are entered under in the *Criteria* cell below the field name in the query grid.

b) Click to position the insert point in the *Criteria* cell for the field to be used for your record selection criteria.

c) Type the expression to be used to select records (See the tables on the previous pages for examples of how to enter valid criteria).

- If *two* conditions must be met in order for a record to be selected, criteria for the second condition must also be specified in the appropriate field in the grid. Repeat steps b) and c) above to create the second condition.
- If any of a number of conditions could be met to select a record, add additional criteria in the OR cells below the criteria cell, or type the word "or" between criteria.

d) To sort the resulting records using the selected field, select *Ascending* or *Descending* in the *Sort* cell in the grid.

Exercise 1 - Creating Record Selection Criteria

In this exercise you will create record selection criteria for an existing query, and you will save the revised query under a new name.

1 Open the query entitled *ISS_Q1 – ISIS Opening Information* in design view.

2 Run the query. Notice that there are 667 records included in the output.

3 Return to the query design screen then use the scroll bar underneath the query grid to display the *OPEN_CATEGORY_CD* field.

4 Click in the *Criteria* cell in the *OPEN_CATEGORY_CD* field, then change the criteria to:
="FTSBF"
Your grid should match the grid shown below:

Field:	OPEN_CATEGORY	PRESCRIPTN_STS	SILV_FUND_SRCE	RESPONSIBILITY_
Table:	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening
Sort:				
Show:	☑	☑	☑	☑
Criteria:	="FTSBF"			
or:				

5 Run the query. Notice that only 51 records are extracted, and they are all FTSBF type openings.

6 Choose **File|Save As/Export...** and name the revised query *Q – FTSBF Type Openings*.

7 Close the query and return to the database window.

Exercise 2 - Creating Record Selection Criteria

1 Open the *ISS_Q1 - ISIS Opening Information* in design view.

2 First you will select the openings which were logged. Place the insert point in the *Criteria* cell in the *DISTURBANCE_ST* field in the grid.

3 Enter the following criterion:
="L".
Your grid should match the grid shown below:

Field:	CUT_BLOCK_ID	CLIENT_NAME	DISTURBANCE_D#	DISTURBANCE_ST
Table:	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening
Sort:				
Show:	☑	☑	☑	☑
Criteria:				="L"
or:				

4 Run the query and scroll through the table. Notice that there are now 645 records displayed.

5 Return to the query design window.

You will now add a range of years to the query to show the openings that were logged between Fiscal Years 1986 and 1995

6 Click to place the insert point in the *Criteria* cell in the *DISTURBANCE_DATE* field. Enter the following criterion:
Between #1985/04/01# And #1995/03/31#
(***Note**: You can zoom the criteria cell by pressing <Shift><F2> with the cursor in the criteria cell, or you can widen the column in the grid by dragging on the right hand gridline at the top to make entering the criteria easier*).
Your grid should match the grid shown below:

Field:	DISTURBANCE_DATE	DISTURBANCE_ST
Table:	SIL - Opening	SIL - Opening
Sort:	Descending	Descending
Show:	☑	☑
Criteria:	Between #1985/04/01# And #1995/03/31#	="L"
or:		

7 Set the sort order for the *DISTURBANCE_DATE* field to *Descending*.

8 Run the query. Notice that the records are sorted in descending order by disturbance date, and that 206 records are displayed.

9 Return to the query design screen.

You will now add two fields to the query grid so that you can further customize your record selection criteria.

10 Make sure that you can see the *OPENING_GROSS_AREA* field in the query grid. You will be adding two new fields at this position.

11 Select the *PREV_TREE_SPP1_ST* field in the table in the upper part of the window and drag it into the grid. Drop the field onto the *OPENING_GROSS_AREA* field. Access will insert the new field into the grid and move the *OPENING_GROSS_AREA* field to the right to accommodate it.

12 Drag the *PREV_AGE_CLASS_CD* field and drop it into the grid on to the *OPENING_GROSS_AREA* field. Your grid should now match the screen shown below:

Field:	DISTURBANCE_DAT	DISTURBANCE_ST	PREV_TREE_SPP	PREV_AGE_CLAS	OPENING_GROS
Table:	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening
Sort:	Descending				
Show:	☑	☑	☑	☑	☑
Criteria:	Between #1985/04/01#.	="L"			
or:					

13 Modify the query to display logged openings for which the previous timber type 1 was PLI and the previous Age Class was greater than 5 over the same period. Your grid should match the grid shown below:

Field:	DISTURBANCE_DAT	DISTURBANCE_ST	PREV_TREE_SPP	PREV_AGE_CLASS
Table:	SIL - Opening	SIL - Opening	SIL - Opening	SIL - Opening
Sort:	Descending			
Show:	☑	☑	☑	☑
Criteria:	Between #1985/04/01#.	="L"	"PLI"	>"5"
or:				

14 Run the query. 13 records should be selected.

15 Modify the query to display the records for the same period for which the previous timber type 1 was SX and the previous Age Class was less than or equal to 9. You should get 5 records as a result

16 Exit to the database window and do not save changes.

Multi-Table Queries

Many of the questions that you want to ask cannot be based on a single table because the main subject areas within the SIA database are stored in separate tables.

Although most of the key tables have already been linked via the surrogate keys, you will have to create linkages for certain queries of your own when you develop them. Creating your own multi-table queries is covered in detail later in this guide.

Note: *Tables which have already been linked will only produced rows where data exist in both tables. This is a standard type of join. For examples, not all openings may have ATUs associated with it. Therefore, a query having the opening and ATU table with a inner join will only provide an output showing openings with ATUs. The openings without any ATUs will not show in the output.*

To look for missing information, we will use different join types that will be discussed later.

The SIA database already has some pre-defined multi-table queries which you can use as is, or modify by adding new fields and/or new criteria.

These queries can be listed by clicking the *Query* button in the database window. Once the list of queries has been displayed you can open a query in datasheet view (to view the dynaset of records) or in design view (to view the query grid).

Exercise - Viewing Multi-Table Sample Queries

1 Display and maximize the training database window.

2 Click the *Query* button to display a list of all of the sample queries.

3 Select the query entitled *ISS_Q1 - Forest Cover*.

4 Choose *Design* to display the query grid. Notice that there are two tables included in this query, and they are linked via the *OPENING_SKEY* field. The link is a *one to many* relationship, since one opening can have many different forest cover polygons/layers.

5 Pan the window left to right using the scroll bar so that you can examine the query grid. Notice which fields have been pulled from the *ISS - Forest Cover* table and which have been pulled from the *SIL - Opening* table.

6 Click the *Datasheet* button (or the *Run* button) to run the query. Examine the contents of the output.

7 Press <Ctrl><F4> to close the query and return to the database window.

Modifying the Sample Multi-Table Queries

The sample queries can be modified and resaved, or saved under a new name as a new query. This method can save you time if the sample query has most of the information that you need.

To modify a query, you simply add or rearrange the fields, and add any additional criteria or sorts.

Exercise 1 - Modifying Sample Queries

1 Open the query entitled *ISS_Q1 - Openings Not FG with FC and SU info* in design view.

2 Click in the *Sort* cell in the *OPENING* field, and choose *Ascending*.

3 Run the query. Notice that some openings have several records and the records are grouped by Opening.

4 Return to the query grid, and change the sort criteria to *Not Sorted* in the *OPENING* field. Add a descending order sort by *SILV_POLYGON_AREA*.

5 Run the query. Notice that the records are now sorted by *SILV_POLYGON_AREA* in descending order.

6 Return to the query grid.

7 Click in the *Criteria* cell in the *DISTURBANCE_DATE* field. Type the following criteria for this cell:
Between #1985/04/01# And #1995/03/31#

8 Run the query. Notice that only the records between the specified dates are extracted. There should be 1020 records in the output.

9 Change the sort so that the records are displayed in descending order by disturbance date. Modify the criteria to display disturbance dates between June 1st, 1989 and June 1st, 1994.

10 Run the query to test your new criteria. Notice that 309 records are extracted.

11 Close the query without saving changes.

Exercise 2 - Modifying Sample Queries

1 Reopen the *ISS_Q1- Openings Not FG with FC and SU Info* query in design view.

2 Use the query grid to find out how many records match the following criteria:

a) Treatable Area does not equal 0 (zero) _____

b) Treatable Area does not equal 0 (zero) and the Maximum Density is greater than or equal to 2500 _____

c) Treatable Area is greater than or equal to 40 and the Maximum Density is greater than or equal to 2500 _____

d) Cancel the criteria used in c), then find records for which the Disturbance date was on or after April 1st, 1993. Sort the data in descending order of Disturbance Date _____

3 Return to the query grid and close the query without saving changes.

Creating One-Table Queries

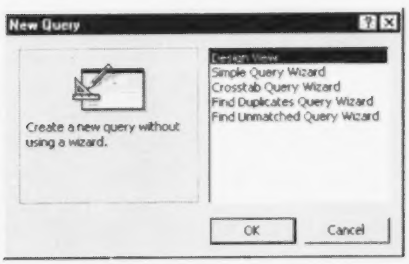
Introduction

When there is no sample query available you will need to create a query from scratch.

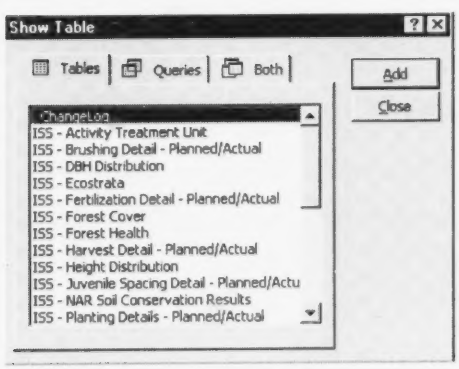
Creating a select query with one table is relatively straight forward. The procedure is as follows:

a) From the *Database* window, click the *Query* object button to display the query list.

b) Choose *New*. Access will present the *New Query* dialog box shown below. Choose *Design View* from this dialog box to access the Query Design window.



c) Access will present the *Show Table* dialog box. Double-click the table that the new query will be based on to add it to the Query window, then choose *Close* to close the *Add Table* dialog box.



d) Make sure that the *Select Query* button is the active query type, then modify the query window by dragging the split bar down to make more room for the table box. Adjust the table box size so that you can see the field names.

e) Add the fields required in the query output to the query grid. Fields should be added in the order that they are to appear in the output, and in the correct order if a hierarchical sort is required (the fields used in a sort are used in order from left to right). ***Note**: You can drag a field to a new position in the grid if you misplace it - Select the field, then use drag and drop to relocate it or use cut and paste from the edit menu).*
To add fields do any one of the following:

- Drag the field from the table box to the required column in the grid
- Double-click the field in the table box to add it to the next available column in the grid
- Click the pop-up button in the field cell in the query grid and select the required field.

f) Add any criteria required to the criteria cell(s) in the appropriate column(s) in the grid.

g) Add any sort criteria to the sort cell(s) in the appropriate column(s) in the grid.

h) Run the query to make sure that the output is what you expect. (*Hint: If you do not receive any output, simplify your criteria and look at the data format. You may have entered your criteria in a different format than it is stored*).

i) Use **File|Save As/Export...** to name and save your query.

Exercise - Creating a One-Table Select Query

Create a Query to List the Opening Information for the District

1 From the query panel of the database window choose *New*.

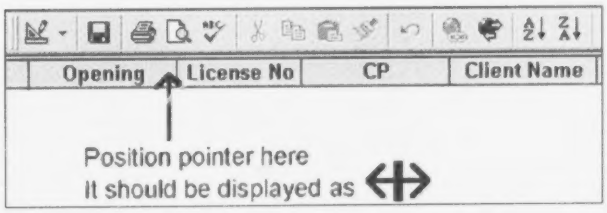
2 Choose *Design View* from the *New Query* window.

3 Double-click the *SIL - Opening* table from the *Show Table* window, then choose *Close*.

4 Drag the split bar in the query window so that it is about 2/3rds of the way down. Widen and lengthen the *Opening* table box.

5 Add the following fields to the query grid using one of the methods listed in step e) on the previous page.
OPENING
FOREST_FILE_ID
CLIENT_NUMBER
CLIENT_NAME
OPENING_LOCN_NAME
OPEN_CATEGORY_CD
OPENING_GROSS_AREA
DISTURBANCE_ST
DISTURBANCE_DATE

6 Run the Query. In the resulting table you can customize the column widths for the output by dragging on the separator bar between each field name. Position the mouse pointer on the separator bar to obtain a two-headed arrow pointer than drag to narrow or widen the column to the left. You can also double-click on the separator bar for "best fit".



Use these techniques to adjust the column widths so that you can see all of the output fields, then return to design view.

7 Add an ascending order sort to the Opening field and rerun the query.

8 Use the appropriate criteria cells to limit the output to Openings which were logged in fiscal 1986 (i.e. between 1985/04/01 and 1986/03/31 inclusive.
DISTURBANCE_ST ="L"
DISTURBANCE_DATE Between #1985/04/01# and #1986/03/31#

9 Run the query. 24 records should be included in the output.

10 Modify the sort criteria to sort by *DISTURBANCE_DATE* in descending order. (i.e. Select *Not Sorted* in the *OPENING* field, then add the new sort to the *DISTURBANCE_DATE* field.

11 Run the query then return to the query grid and choose **File|Save As/Export...**.

12 Name the query: *Q - Logged Openings 1986 Fiscal*

13 Close the query window.

Exercise - One-Table Queries

Create a Query to List the Planned Activities for the District

1 Create a new query based on the *Activity Treatment Unit* table.

2 Add the following fields:
OPENING
FOREST_FILE_ID
CLIENT_NUMBER
SILV_BASE_CODE
PLND_TECHNIQUE_CD
PLND_METHOD_CODE
RESULTS_IND
PLND_DATE

3 Add sort criteria to sort the data in ascending order by base, technique and method.

4 Select activities planned in 1995
PLND_DATE BETWEEN #1995/01/01# AND #1995/12/31#
RESULTS_IND = "N"

5 Run the query. 17 records should be included in the output. Save this query as: *Q - Planned ATUS in 1995* and close the Query window.

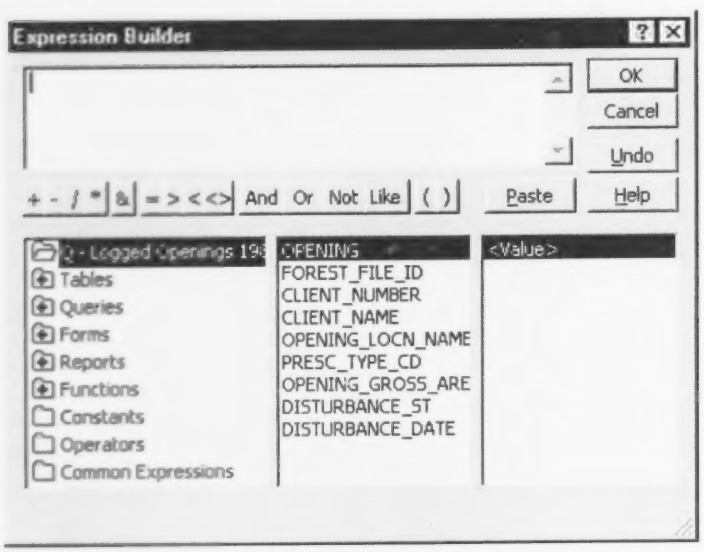
Adding Calculated Fields to a Query

Some queries call for field results which are not actually stored in the SIA database. Calculations such as the Estimated Cost per Hectare for a Site Preparation or Planting Treatment, the Net Area to be Reforested, and the Planned and Actual Costs of Planting per Hectare are examples of results that you might want to be able to generate. In addition, you may want to combine two fields into one to obtain a more readable result. These types of results can be obtained by building calculated (virtual) fields as part of your query output.

Calculated fields are created in a query by manipulating existing fields within SIA using customized expressions which act on those fields.

The table on the following page lists some of the many Mathematical and String Operators which can be used to build expressions within an Access query. There are also a whole range of functions which are available for more sophisticated expressions.

The Expression Builder dialog box (shown below) can be used to assist you in building complex expressions. The expression builder can be displayed by clicking the expression builder button or by pressing <Ctrl><F2>. **If you select the fields you want to use in your calculation and then save your query prior to going into the Expression Builder, they are easier to call up.**



Operator	Definition	Example Expression	Resulting Output
+	Add	[*TREES_PLANTED1*]+[*TREES_PLANTED2*]	1200+550=1750
-	Subtract	*[PLND_TRTMNT_AMT]-[ACT_TRTMNT_AMT]*	25-20=5
*	Multiply	[COST/HA]*[#Hectares]	60*3=180
/	Divide	[TOTAL_COST]/[#Hectares]	180/3=60
Left	Truncate from left at specified position	Left([*TREE_SPECIES1_CD*],*2*)	PLI=PL
Right	Truncate from right at specified position	Right([*OPENING*],*4*)	82F03000__50=50
Mid	Extract text starting at specified position, and returning specified number of characters	Mid([*ATU_COMPLETION_DT*],6,2)	1995-03-31=03
&	Join two fields (Concatenate)	[BGC_ZONE]&[BGC_SUBZONE]	ESSFwc
Year	Returns the Year of a date argument	Year ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 1995
Month	Returns the Month of a date argument	Month ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 9
Day	Returns the Day of a date argument	Day ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 27
FiscalDt	Returns the fiscal year of a date argument	FiscalDt([ATU_COMPLETION_DT])	ATU Comp Date = 1995-09-27 Result = 1996
If	Returns one of two arguments based on an expression	If([[REFERENCE_YEAR]=1995),[AVG_AGE],(1995-[REFERENCE_YEAR]))	

***Note**: When you build an expression based on fields within the SIA database, you must enclose the field name in square brackets, and match the name exactly as it appears in SIA. Functions or calculations which refer to fields are also case-sensitive.*

System Generated Fields

There are a number of system generated (calculated) fields displayed within ISIS which are not actually stored. The following list shows you some of these fields and the method used to generate the same output using a calculated field in an SIA query:

Field	**Calculation in SIA**
Planted Stems/Ha	[ACT_PLANTED_NO]/[ACT_TRTMNT_AMT]
Planned Cost/Ha	[PLND_TOTAL_COST]/[PLND_TRTMNT_AMT]
Actual Cost/Ha	[ACT_TOTAL_COST]/[ACT_TRTMNT_AMT]
Planning PHSP Minimum WSS/Ha	Use MIN_STOCKING from Standards Unit

Exercise - Creating Calculated Fields

Determine the per hectare cost of planned activities. Note, this exercise is based on an exercise on Page 61.

1 From the query panel of the database window select the query *Q - Planned ATUs in 1995.*

2 Choose *Design* to display the query grid for this query.

3 Add the following two fields to the query:
PLND_TRTMNT_AMT
PLND_TOTAL_COST

** *save your query*

4 Use the horizontal scroll bar to display the first empty column in the grid.

5 Click to place the cursor in the field cell in this column.

6 Click the right mouse button to display the short-cut menu and choose *Build.*

7 Build the following expression to calculate the per hectare cost of the planned activity:
[PLND_TOTAL_COST]/[PLND_TRTMNT_AMT]. Hit <Enter>

8 Run the query and study the output. Notice that the title of the new column is *Expr1,* and that the values in this column are not formatted.

9 Return to the query grid and leave the query open for the next exercise.

Field Properties

Each field in an Access table has a number of properties attached to it. These properties are defined during the table design phase. Some properties which can be defined for a field include: the field size, data format, the number of decimal places (if applicable), and a caption (column heading).

The calculated field in the previous exercise did not have a particularly useful column heading, and the data in the column was not formatted appropriately, however, you can also modify the properties of any field in the query grid.

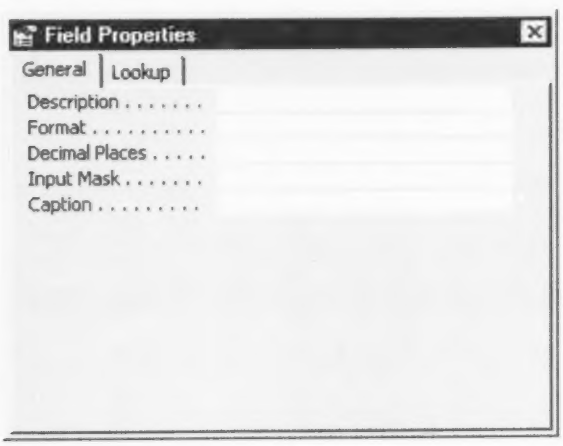
To modify the properties of a field:

a) Click to place the insert point in the field cell then do any one of the following:

- Choose **View|Properties**
- Click the *Properties* button on the toolbar
-
- Right-click in the *Field* cell and choose *Properties* from the short-cut menu

b) Access will display the *Field Properties* dialog box (shown below).

d) Customize the required properties, then double-click the *Field Properties* control box to return to the query grid.



***Note**: The Format and Decimal Places text boxes provide pop-up buttons which allow you to select from a list of available options for the particular property.*

Exercise - Customizing Field Properties

1 The *Q - Planned ATUs in 1995* query should be active and displayed in design view.

2 Select the *Field* cell in the *Expr1* column.

3 Right-click to display the short-cut menu and choose *Properties*.

4 Click in the *Format* text box to obtain the pop-up button.

5 Open the Formats list box and select *Currency*.

6 Press <Tab> to move the insert point to the *Decimal Places* box and specify 2 decimal places.

7 Use <Tab> to move to the *Caption* text box and enter the following caption:
Cost/Ha

8 Choose *Close* from the *Field Properties* control menu (or press <Alt><F4>).

9 Run the query and notice the changes to the output.

10 Return to the query grid, save and close your query.

Exercise - Creating Other Queries with Calculated Fields

The following query will calculate the estimated total cost for site preparations and planting treatments for treatment units which the forest service is responsible for. Estimated cost will be based on $450.00 per hectare.

1 From the database window, create a new query based on the *Activity Treatment Unit* table.

2 Add the following fields to the query:
OPENING
FOREST_FILE_ID
ACTIVITY_TU_ID
RESPONSIBILITY_CD
SILV_BASE_CODE
PLND_TRTMNT_AMT
RESULTS_IND

Save your query as 'Q - Cost of Activities'

3 Highlight the field column to the right of the *PLND_TRTMNT_AMT*. Press <Insert> to create a new column.

4 Create the following expression using the Expression Builder.
*[PLND_TRTMNT_AMT] * 450*

5 Format this field as *Currency*, and add the following caption
Est Cost

6 Add the following criteria to the *SILV_BASE_CODE* criteria cell to limit the output to Planting and Site Preparation.
IN ("PL","SP")

7 Add the following criteria to the *RESPONSIBILITY_CD* criteria cell to limit the output to ATU's for which responsibility lies with the Forest Service.
="F"

8 Add the following criteria to the *RESULTS_IND* criteria cell to limit output to planned ATU's
="N"

9 Sort the output by *SILV_BASE_CODE* in descending order, then by *PLND_TRTMNT_AMT* in ascending order.

10 Add the caption: "*Treated Area*" to the *PLND_TRTMNT_AMT* field then run the Query. You should have 26 records included in the output.

11 Modify the column widths in the datasheet screen, then return to the design screen to correct any errors.

12 Save the query as: *Q - Cost of Activities and close query window.*

Optional Exercise #1 - Calculated Fields

You have been asked to provide a list of all of the ATU's for which planting activities were completed in the 1994 fiscal year. Your report should also include a column which shows the planted density for each ATU. The list should be displayed in order by opening number.

1 Create a new query based on the *Planting Details- Planned/ Actual* table.

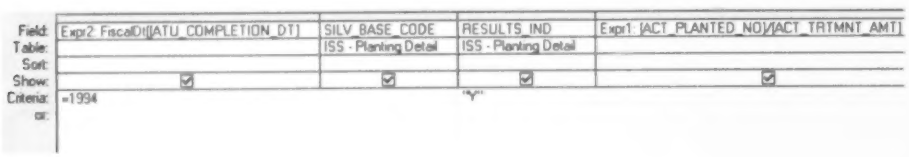
2 You will need the following fields to build your query: *OPENING, FOREST_FILE_ID, ACTIVITY_TU_ID, ATU_COMPLETION_DT, SILV_BASE_CODE,* and *RESULTS_IND.*

3 You need to create a calculated field to provide the planted density. Use the *ACT_PLANTED_NO* and the *ACT_TRTMNT_AMT* fields to build this calculation. This field should be given the column heading "*Planted Density*" and should be shown in Fixed format.

4 Change the ATU_Completion_DT to fiscal year using the FiscalDt function and limit to 1994. Limit the output to records where the Results Indicator flag has been set up to "Y".

5 Add the opening field sort.

6 Run the query. Your output should include 32 records. Correct any errors then save the query as: *Q - Planted SPH - 1994.*
(The key part of the query grid is shown below for reference):



Field:	Expr2: FiscalDt([ATU_COMPLETION_DT]	SILV_BASE_CODE	RESULTS_IND	Expr1: [ACT_PLANTED_NO]/[ACT_TRTMNT_AMT]
Table:		ISS - Planting Detail	ISS - Planting Detail	
Sort:				
Show:	☑	☑	☑	☑
Criteria:	=1994		"Y"	
or:				

7 Close the query window.

Parameter Queries

Parameter Queries are very useful for expanding the functionality of a query. For queries that you run often, you can automate the process of changing the criteria by creating a parameter query. The parameter query prompts you for the criteria when you run the query, thus eliminating the necessity of opening the query in Design view and changing the criteria.

To create a parameter query:

a) Create a select query.

b) In the Criteria row for the field in which you usually enter criteria, type the prompt you want to appear in the dialog box, enclosed in square brackets with the appropriate comparison operator

Here are some examples:

A. =[Enter a specie code]

B. **Between** [Enter start date] **and** [Enter end date]

C. **Like** [Enter Base Code]&"*"

(This will give everything if you leave Base Code blank)

Exercise - Creating a Parameter Query

1 Create a new query based on the *ISS - Forest Cover* table.

2 Add the following fields to the query:

OPENING
FOREST_FILE_ID
RESPONSIBILITY_CD
SILV_POLYGON_NO
FOR_COVER_LAYER_CD
SILV_POLYGON_SUFIX
FOR_COVER_RANK_CD
SILV_POLYGON_AREA
TREE_SPECIES1_CD
TREE_SPECIES2_CD

3 Sort the output in ascending order by *OPENING* and then in descending order by *SILV_POLYGON_AREA*.

4 Click in the Criteria cell in the *TREE_SPECIES1_CD* field. Type the following parameter for this cell:
[Enter Species 1]

5 Run the query. When prompted for the Species 1 Code enter SX

6 Notice that only the records for which SX is the primary species are extracted.

7 Add the following parameter in the criteria cell for the *TREE_SPECIES2_CD* field and rerun the query using SX and BL as the primary and secondary species.
[Enter Species 2]

8 Use the parameter query to find out how many records match the following criteria:

a) Primary species 1 is "PLI" and Secondary species is "SX" _____

b) Primary species is "SX" and Secondary species is "PLI" _____

c) Primary species is "SE" and Secondary species is "BL" _____

9 Return to the query grid and choose **File|Save As.**

10 Name this new query:
Q - FC - Species 1 and 2

11 Close the query window.

Summary Queries

There are two *basic* types of select queries: detail queries and summary queries.

Detail queries produce lists of records (dynasets) which meet specific query criteria that you specify. Summary queries provide a synthesis of these detailed reports by collapsing the details into a more compact format. For example, a detail query would be used to list all of the ATU's which had juvenile spacing over the past year, whereas a summary report would be used to calculate the total number of hectares which were juvenile spaced.

You can use a query to calculate totals for all records or for groups of records. For example, you can use a summary query to answer questions such as: What is the *total* area which has not been satisfactorily restocked? or What is the total area treated *by activity* for all activities completed during fiscal year 1993?

Creating a Summary Query

A summary query is created in much the same way as a detail query, however a *Totals* row must be added to the query grid in order to specify the required summary statistic.

The *Totals* row is added to the grid by clicking the *Totals* button on the query toolbar, or by choosing **View|Totals**. Once the *Totals* row has been added to the query grid, you can specify the calculation required by using one of the summary operators in the column (field) which you want to summarize on.

The Group By Operator

When you add the Totals row to the grid, Access automatically includes the *Group By* operator in each of the Totals cells.

The Group By operator collapses the detail in each of the columns containing the operator. For example, if several records have the same Base Code, when the data is grouped by the Base Code field, each Base Code will only appear once in the output. If *Group By* appears in another Total cell in the grid, then the data will be grouped on that column also.

Group By is treated in a hierarchy reading from left to right, therefore fields should be added to the query grid in the order that they are to be grouped by. ***Note**: You can drag a field to a new position in the grid if you misplace it - Select the field, then use drag and drop to relocate it or use cut and paste from the edit menu).*

Warning: If you are limiting your output using criteria (i.e. within a year range) it is a good idea to preview the data in detailed format first to ensure you are pulling the correct information. Once you have the data in summary type format there is no way to confirm that the data being rolled up is correct.

The following exercise will demonstrate how the *Group By* function works.

Exercise - Using the Group By Operator

1 From the database window, create a new query based on the ISIS *Forest Cover* table.

2 Add the *TREE_SPECIES1_CD* field to the query and run the query. Notice that there are 2830 records in the table.

Σ 3 Return to the query design screen and choose **View|Totals** (or click the *Totals* button on the toolbar). Notice the addition of the totals cell, and the *Group By* operator.

4 Run the query and notice that there are now only 19 records included in the output. These represent the 18 unique species of trees recorded in the table (plus one record for those records in the table which do not have a value in the *TREE_SPECIES1_CD* field).

5 Exit the query design without saving.

Other Summary Operators

Access provides a number of other *Summary Operators* for use with fields that have a numeric value. The table below lists the most common of them with a description of what they do.

Operator	Definition
Sum	Sums the values based on the right-most group by field
Avg	Averages the values based on the right-most group by field
Min	Returns the smallest value based on the right-most group by field
Max	Returns the largest value based on the right-most group by field
Count	Counts the number of occurrences based on the right-most group by field
StDev	Returns the Standard Deviation based on the right-most group by field
Var	Returns the Variance based on the right-most group by field
Where	Allows you to specify criteria to limit the records before a calculation is performed.

Exercise - Creating a Simple Totals Query

Determine the Total NSR Area for the District.

1 Create a new query using the *SIL -Opening* table.

2 Add the *NSR_TOT_AREA* field to the query and set the *Sort* order to *Descending.*

3 Run the query and scroll through the results.

Σ

4 Return to the query design screen, then choose **View|Totals** (or click the *Totals* button on the toolbar). Notice that a *Totals* row has been added to the grid with the *Group By* operator automatically entered into the *Total* cell. Change the Group By operator to Sum by clicking in the Total cell and selecting Sum from the resulting pop-up list.

5 Run the query again. Notice that only one record is included in the output and the output column name has been changed to: *SumOfNSR_TOT_AREA*. This record represents the Total of all of the values in the *NSR_TOT_AREA* field.

6 Return to the query design screen and save the query as: *Q - District NSR.*

7 Close query window.

Obtaining Sub-Totals

You can obtain sub-totals for specific groups of records by including the field that you want to sub-total on, and specifying the Group By operator for that field.

Exercise - Using Sub-Totals in a Summary

In the following exercise you will find the NSR totals for each licence within the district.

1 Open the *Q - District NSR* query in design view.

2 add the *FOREST_FILE_ID* field to the query grid.

3 Specify *Not Sorted* for the *NSR_TOT_AREA* field, and an ascending order sort for the *FOREST_FILE_ID* field.

4 run the query. Notice that some of the records show zero as the *SumOfNSR_TOT_AREA*. You need to add criteria to exclude the records for which *NSR_TOT_AREA* field contains a zero.

5 Add the following criteria to the *NSR_TOT_AREA* field:
<>0

6 Run the query and scroll through the output. Note that the first record shows no License Number.

7 Return to the query design screen and enter the following criteria for the *FOREST_FILE_ID* field:
<>""

8 Run the query and view the output.

9 Return to the query design screen and save the query as: *Q - District NSR By License*

Optional Exercise

List total Silviculture Activity area completed by major licensees by year complete and base code.

Hints:

Use the ISS - Activity Treatment Unit table

Use the OPEN_CATEGORY_CD "FTML"

Group by Year([ATU_COMPLETION_DATE]) and SILV_BASE_CODE.

Sum on ACT_TRTMNT_AMT

You should get 66 records in your output.

Close query window without saving the query.

Specifying Criteria for Totals Fields

There are 2 ways to specify criteria for totals fields:

1. *Calculate totals and then limit output* to totals meeting specified criteria.

   For example: Calculate total brushing cost by funding source and then limit output to those >$100,000

2. Specify criteria that *limit records before calculating* the total (use the Where clause).

   For example: Calculate the total brushing cost by funding source if the cost/atu was >$30,000.

The first method is implemented by simply entering your criteria in the criteria row below the field being totalled.

The second method requires the *Where* operator (see next section).

The Where Operator

The *Where* Operator allows you to specify criteria to limit the records pulled before a total is calculated.

Exercise - Using The Where Operator

In this exercise you will create a summary query which shows the total number of trees planted by year (from ISIS). You will then modify the query to see just the number of trees planted since 1990.

1 Create a new query based on the *Planting Details – Planned/Actual* Table.

2 Add the following fields to the query grid:
ATU_COMPLETION_DT
ACT_PLANTED_NO

3 Add the *Totals* cell to the query grid.

4 Set up the calculation as follows:
Group By - Year([ATU_COMPLETION_DT])
Sum - ACT_PLANTED_N

5 Run the query. The query should calculate the number of trees planted for each year. There should be 25 rows.

6 Return to the query design screen and enter criteria to select years >1990. Run your query. There should be 9 rows.

7 Return to the query design screen and change the *group by* clause under ATU_COMPLETION_DT to the *where* clause. Run the query. There will be one row returned and the total number of trees planted within that year range (>1990) should be 7,213,613.

8 Close query window without saving query.

Cross Tabulations

A specialized type of summary query is called a cross-tabulation.

A cross-tabulation typically uses three fields to produce a table of cross-tabulated values. One field supplies the row headings, another field supplies the column headings, and a third field supplies the values to be cross-tabulated. Data summarized in this row by column format produces a more compact summary than a select query.

The procedure to produce a crosstab query is as follows:

a) Create a new query.

b) In the query design window, choose **Query|Crosstab Query** (or click the *Crosstab Query* button on the toolbar). Access will automatically add *Total* and *Crosstab* rows to the query grid.

c) Specify the fields, sorts and record selection criteria as you would for creating a select query.

d) Select the *Crosstab* cell in the field whose values you want to appear as rows, and select *Row Heading* from the drop-down list box. You must leave the default *Group By* operator in the *Total* cell for this field.

e) Select the *Crosstab* cell in the field whose values you want to appear as columns, and select *Column Heading* from the drop-down list box. You must leave the default *Group By* operator in the *Total* cell for this field.

f) Select the *Crosstab* cell in the field whose values you want to use in the cross-tabulation, and select *Value*. In the *Total* row for this field, select the type of aggregate function you want for the cross-tabulation (e.g. Sum, Avg, or Count).

g) If record selection criteria are to be used, select *Where* in the *Total* cell for the desired field, and enter an expression. Leave the *Criteria* cell blank for Value fields.

Exercise 1 - Creating Cross-Tabulations

In the following exercise, you will create a cross-tabulation to summarize the Total Treated Area for each Silviculture Base Code by Funding Source.

1 Create a new query based on the *Activity Treatment Unit* Table.

2 Choose **Query|Crosstab Query** (or click the *Crosstab Query* button on the toolbar). Notice that Access automatically adds the *Total* row and the *Crosstab* row to the query grid.

3 Add the following fields to the query grid:
ACT_FUND_SRCE_CD
SILV_BASE_CODE
ACT_TRTMNT_AMT
RESULTS_IND

4 In the *Crosstab* cell under *ACT_FUND_SRCE_CD*, select *Row Heading*.

5 In the *Crosstab* cell under *SILV_BASE_CODE*, select *Column Heading*.

6 In the *Crosstab* cell under *ACT_TRTMNT_AMT*, select *Value*.

7 Leave the *Total* cell set to *Group By* for the ACT_FUND_SRCE_CD and *SILV_BASE_CODE* fields, but set the *Total* cell to *Sum* for the *ACT_TRTMNT_AMT* field. Set the criteria cell for the *RESULTS_IND* field to ="Y"

8 Run the query and study the output. Set the column widths in the output to the *Best Fit* option. (Select the columns you can see on the screen and choose **Format|Column Width|***Best Fit.*, move across the screen and repeat)

9 Save the query as: *Q - Treated Area by Fund/Base Code*

10 *Close the query window.*

Creating Multi-Table Queries

Introduction

On many occasions you will need to draw information from more than one subject area to answer a specific information need. For example, you may want to obtain a list of all planting activities with a request id entered, or you may want to generate a list of all the openings, which were harvested, using the clear-cut method over a particular range of years. To obtain this type of information you will need to base your query on two or more tables in order to retrieve the relevant data.

Care and planning is required when using two or more tables as the basis for a query in order to ensure that the results are accurate and also to make sure that the information is retrieved in the most efficient way.

Access makes the process of creating multi-table queries relatively straightforward because the relationships (joins) between critical tables can be set up in advance, and recorded as part of the basic database design. This has already been done for you in the SIA database, thus most of the necessary table linkages are already in place.

The SIA tables are generally linked using the surrogate keys (S_KEY fields) in each table, since this provides the most efficient method of retrieving data from multiple tables using a query.

Select, Action, Summary, Cross-Tabulation and Parameter queries can all be performed using more than one table as the basis of the query.

Creating a Two Table Query

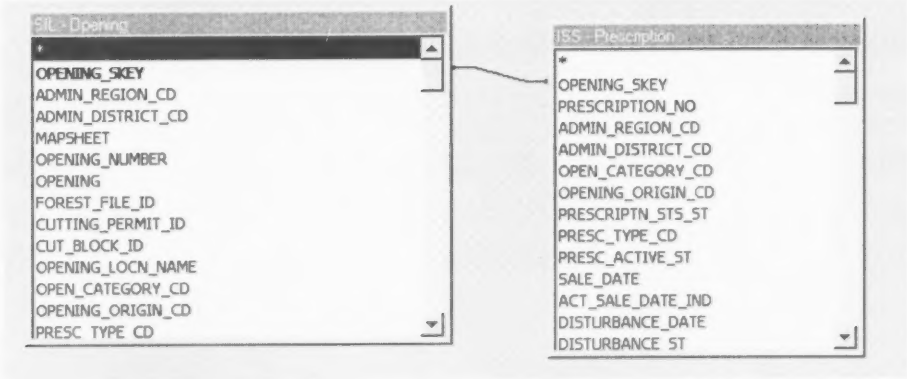
The procedure for creating a two table query is as follows:

a) From the *Database* window, click the *Query* object button to display the query list and choose *New* (or choose **Insert| Query**).

b) Access will present the *New Query* dialog box. Choose *Design View* from this dialog box to access the Query Design window.

c) Access will present the *Show Table* dialog box. Select each table required for the query and click *Add* (or double-click each table to be included in the query). Once all tables have been added, choose *Close* to close the *Show Table* dialog box.

d) Make sure that linkages (relationships) exist between the selected tables. The joins should connect the two tables using a Surrogate key (such as *OPENING_SKEY*) which is common to both tables. All surrogate key fields end in _SKEY, and uniquely identify each record in the home table. If no joins exist, select the SKEY in the main table, then drag it over to the same SKEY in the second table. The screen below shows a typical join. Notice that the SKEY is displayed in bold in the table for which it is the primary key.
Note: If a join does not exist between the two tables you have selected, consider what it is that you are really after.



e]Add the fields required in the query output to the query grid. Fields should be added in the order that they are to appear in the output, and in the correct order if a hierarchical sort is required. If the source table is not displayed for each field in the query grid and you want to see them then choose **View|Table Names**

f) Add any criteria required to the criteria cell(s) in the appropriate column(s) in the grid.

g) Add any sort criteria to the sort cell(s) in the appropriate column(s) in the grid.

h) Run the query to make sure that the output is what you expect.

i) Use **File|Save As/Export...** to name and save your query.

Exercise - Creating a Two-Table Query

In this exercise you will create a Query to list the openings which were harvested using the clearcut method after 1985 with opening areas greater than 25 hectares

1 From the query panel of the database window choose *New.*

2 Choose *Design View* from the *New Query* window, then click *OK*

3 Double-click the *SIL - Opening* table from the *Add Table* dialog box, then double-click the *ISS - Prescription Silvicultural Systems* table to add both tables to the query. Choose *Close* to close the *Add Table* dialog box.

4 Arrange the query window so that you can increase the size of the table boxes, but also see the query grid.

5 Study the relationship between the two tables. Notice that the tables are joined using the *OPENING_SKEY* which provides an unique identifier for each opening in the *SIL - Opening* table. The join is a *one-to-many* join indicating that for each unique opening in the opening table there are many corresponding openings in the *ISS - PHSP Harvest* table. In other words, one opening may be matched to several silvicultural systems.

6 Add the following fields from the *SIL - Opening* table to the query grid
OPENING
FOREST_FILE_ID
DISTURBANCE_DATE
OPENING_GROSS_AREA

7 Add the *SILV_SYSTEM_ST* and *SILV_SYSTEM_SKEY* field from the *ISS – Prescription Silvicultural Systems* table to the grid.

8 Enter appropriate criteria in the relevant fields:
SILV_SYSTEM_ST ="CLEAR"
DISTURBANCE_DATE >=#1985/01/01#
OPENING_GROSS_AREA >=25

9 Run the Query. Return to the query design, and enter criteria to sort by disturbance date in descending order.

10 Run the query and review the output. You should get 41 records.

11 Save the query as: *Q - Clearcuts After 1985.*

12 Close the query window.

Joining Tables

If the two tables required for the query have not been permanently joined, you will have to create the join manually.

The procedure is as follows:

a) Create a new query adding the two required tables to the query window.

b) Determine the appropriate SKEY to use for the join and make sure that it is displayed in each of the table boxes.

c) Drag the SKEY field from the table on the left and drop it on the matching SKEY field in the table on the right.

d) Build the rest of the query in the normal way.

Exercise - Creating a Join

In this exercise you will create a query which reports on the Openings which had planted stands between 1975 and 1980. The list will indicate the number of planted stems per hectare, the current total stems per hectare, and the well-spaced stems per hectare for those openings.

1 Create a new query using the *ISS - Planting Details Planned/Actual* and the *ISS - Forest Cover* tables as the basis for the query. Notice that these tables are not permanently joined.

2 Arrange the query window so that you can increase the size of the table boxes, but also see the query grid.

3 You will join these tables using the *OPENING_SKEY* since this is the only key field which is common to both tables. Drag the *OPENING_SKEY* field from the table on the left and drop it on the *OPENING_SKEY* field in the table on the right. A line should now be linking the two tables together.

4 Add the following fields with the following criteria (shown in brackets) from the *ISS - Planting Details Planned/Actual* table:
OPENING (Sort - Ascending)
ACTIVITY_TU_ID
ATU_COMPLETION_DT (Between #1975/01/01# And #1980/12/31#)
ACT_PLANTED_NO (<>0)
PLANTED SPH [ACT_PLANTED_NO]/[ACT_TRTMNT_AMT] (***Note**: This is a calculated field and should be entered exactly as shown into the field cell in the grid - press enter to let Access know you are finished defining it. Set the Properties for this field to Standard Format to 0 decimal places)*
RESULTS_IND (="Y") - Turn "show" off.

5 Add the following fields from the *ISS - Forest Cover* table:
SILV_POLYGON_NO
FOR_COVER_LAYER_CD (="S")
WS_STEMS_PER_HA

6 Run the query. 34 records should be included in the output.

7 Return to the design screen and save the query as:
Q - PL Stems/WS Stems.

8 Close the query window.

More Complex Joins Between Tables

The relationships used between the tables in SIA and in the exercises in this manual have all been set up as normal (inner) joins which specify that only the records which contain exactly matching values in the linked field are included in the output. All openings with ATU's scheduled is an example of exactly matched records.

Some queries will require a more complex join to ensure that all of the records between tables are included in the output. For example you may need to list all openings with or without scheduled ATU's.

Types of Joins

There are three types of joins that can be used between tables:

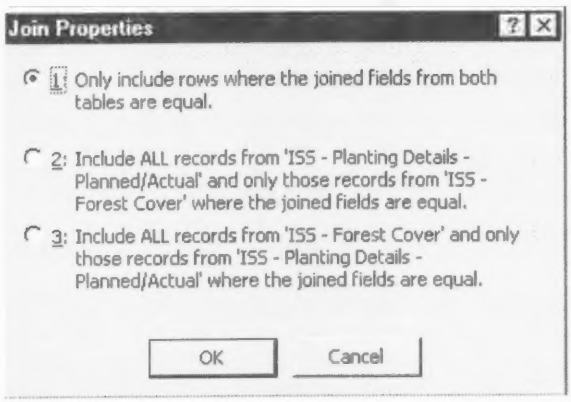
Inner Join: Records from two tables are combined in the dynaset only when there are equal values in the joined fields (this is the default join type).

Left Outer Join: All records from the table on the left of the join are included and only those records from the table on the right which match are included.

Right Outer Join: All records from the table on the right of the join are included and only those records from the table on the left which match are included.

Modifying the Join Type

To modify the type of join between tables, double click the join line. Access will display the *Join Properties* dialog box which is shown on the following page:


Join Properties
1: Only include rows where the joined fields from both tables are equal.
2: Include ALL records from 'ISS - Planting Details - Planned/Actual' and only those records from 'ISS - Forest Cover' where the joined fields are equal.
3: Include ALL records from 'ISS - Forest Cover' and only those records from 'ISS - Planting Details - Planned/Actual' where the joined fields are equal.
OK
Cancel

Exercise - Inner and Outer Joins

In this exercise you will compare the three different join types.

1 Create a new query based on the *ISS - Activity Treatment Unit* and *ISS - Silv Project* tables.

2 You will join these tables using the *SILV_PROJECT_SKEY* since this is the only key field that is common to both tables. Drag the *SILV_PROJECT _SKEY* field from the table on the left and drop it on the *SILV_PROJECT _SKEY* field in the table on the right. A line should now be linking the two tables together.

3 Double click the asterisk in the *ISS - Activity Treatment Unit* table to include all fields in the output.

4 Run the query. How many records are included in the output? _____

5 Return to the query design screen.

6 Double-click the join line between the two tables.

7 Select the Left Outer Join (option 2), then click *OK*.

8 Run the query. How many records are included in the output? _____

9 Return to the query design screen.

10 Double-click the join line between the two tables.

11 Select the Right Outer Join (option 3), then click *OK*.

12 Run the query. How many records are included in the output?

13 Close the query without saving.

Make Table Queries

Occasionally you may want to freeze the results of a query so that you can store the results more permanently. You can use make-table queries to archive old records, to make backup copies of your tables, or to make-tables on which you can base further queries.

Storing the results of a query makes it easier to base a new query on an existing query, and may improve the performance of a very complex query. It eliminates the possibility of the original query changing, and allows you to compile a collection of information. Since a select query produces a dynaset which is based on the current contents of the tables which supply the data, a query based on a query may yield inaccurate results when the database is updated, or if the original query is altered.

Each time you run a make table query it will erase the information from the destination table and copy in the new information. In contrast to a select query, which produces a dynaset or virtual table, *a make table query takes up real disk space* since an actual table is stored as part of the SIA database as a result. You may therefore want to preview the results of a query before you store it in a table.

Warnings on Make-Table Queries

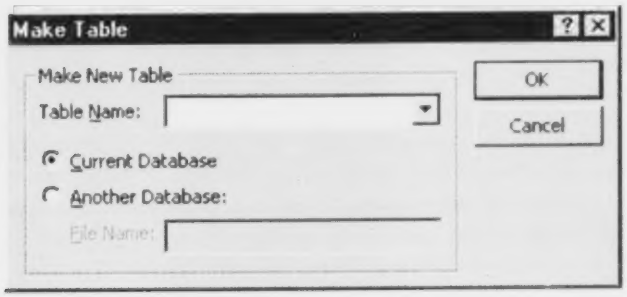
Several factors need to be considered when creating Make-Table queries.

- Creating a new table uses up valuable storage space on the LAN or on your hard disk. Make-Table queries should therefore be used sparingly and should be stored on your hard disk.
- Tables that were created with a Make-Table query are not automatically updated when new data is transferred from Victoria, therefore the query will have to be re-run periodically to make sure that the data stored in it is current.

Creating a Make-Table Query

The procedure for producing a Make-Table query is as follows:

a) Create a new query. In the query window choose **Query|Make Table Query** (or click the *Make-Table Query* button on the toolbar). Access will display the *Make Table* dialog box.

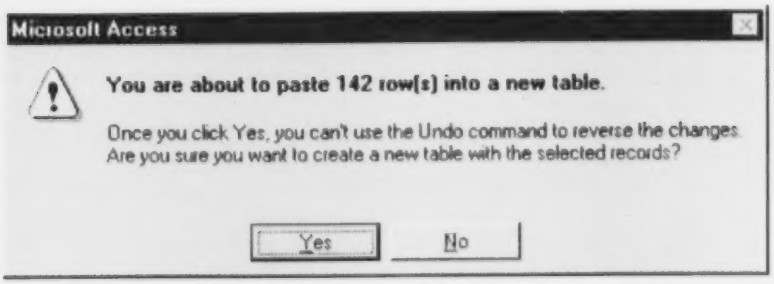


b) Type a name for the new table in the *Table Name* box, or you can click the pop-up button beside the Table Name box and select a table from the list. ***Note**: If the table name you type or select is the same as the name of an existing table, Access will replace the structure and data of the old table with those of the new table.*

c) Choose *OK*. Access will display the *Make Table Query* window.

d) Create the query in the same way that you create a select query by adding tables, fields, sort criteria, and record selection criteria. Define any properties you want for the output fields.

e) Run the query using the *Run* button on the toolbar (or choose **Query|Run**). Access will display a confirmation prompt that tells you how many records will be added to the new table. Choose *OK* to run the query.


Microsoft Access
You are about to paste 142 row(s) into a new table.
Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to create a new table with the selected records?
Yes
No

***Note**: You can preview the query result **without** making the table by clicking the datasheet button. The query result (dynaset) will be displayed but the table will not actually be created.*

Exercise - Making a Table

In this exercise you will create a table that lists the Total Silviculture Accomplishments by License # for fiscal year 1995 for which the Licensee has responsibility.

1 From the database window, create a new query based on the *ISS - Activity Treatment Unit* table.

2 Choose **Query|Make Table Query**

3 In the *Make Table* dialog box, enter the following table name then click *OK*:
MTQ - Accomplishments by License#.

4 Add the following fields in the following order:
FOREST_FILE_ID
ATU_COMPLETION_DT
SILV_BASE_CODE
ACT_TRTMNT_AMT
RESPONSIBILITY_CD
RESULTS_IND

5 Sort the data by *FOREST_FILE_ID*, then by *SILV_BASE_CODE* in ascending order.

6 Add the following criteria to the *ATU_COMPLETION_DT* field:
Between #1994/04/01# And #1995/03/31#

7 Add the following criteria to the *RESPONSIBILITY_CD* field then deselect the *Show* box for this field:
="L"

8 Add the following criteria to the *RESULTS_IND* field then deselect the *Show* box for this field:
="Y"

9 To Preview the result of this query, click the datasheet button. You should get 142 records in the dynaset.

10 Return to the query window and correct any errors if necessary, then click the *Run* button (or choose **Query|Run**). Click *OK* in the Prompt box to complete the transaction.

11 Save the query as: Q - *MTQ - Accomplishments by License#*.

12 Close the query window, then select the *Table* object button in the database window. Notice that your new table is listed.

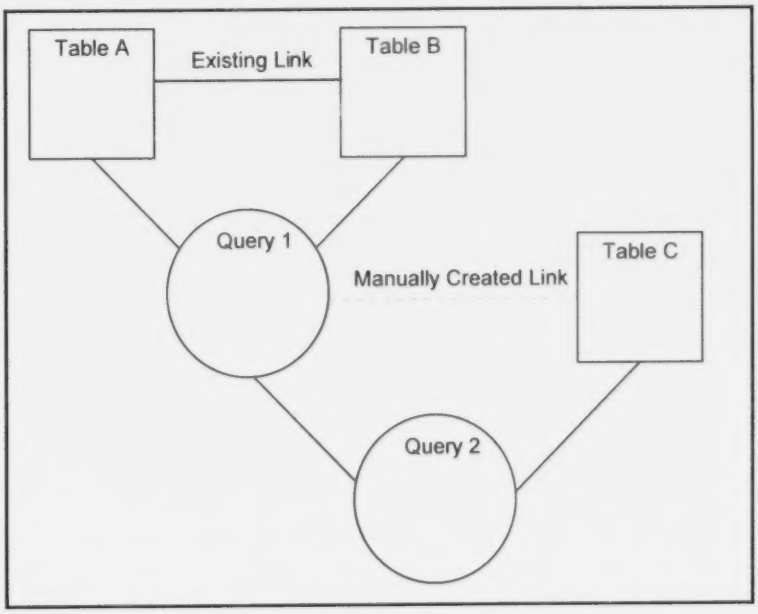
13 Double-click your new table to open it in datasheet view. Close the table and return to the database window, then select the *Query* object button. Notice that your Make-Table query has a different icon to differentiate it from the select queries.

Query on Query

Introduction

On some occasions you will need to draw information from more than one subject area to answer a specific information need. If the query that you are trying to create is particularly complex, it may be an advantage to break the process down into a series of steps which are easier to manage intellectually.

Using a step-by-step approach you can create a query and save it, then link the resulting query to a table to produce another query which gives you the final query result.



When you base a query on another query there are a number of factors to take into consideration:

- Linkages already exist between the main tables in SIA, however you will have to manually create the joins when you link an existing query and a table
- You must make sure that your initial query contains the surrogate keys which will be required to create the join between the query and the table to produce your final query output

Be aware that any changes to the initial query will have an effect on the final query result

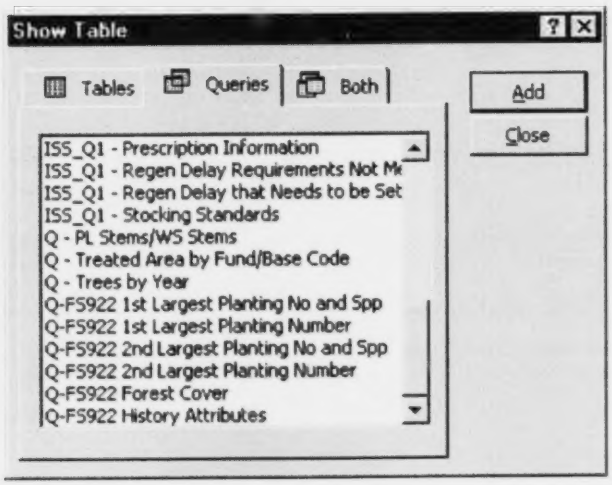
Creating a Query on a Query

The procedure for creating a query on a query is as follows:

a) Create the initial query adding the tables that you need to generate the output. Make sure that any surrogate keys which will be required to join this query to the other table are included in the output.

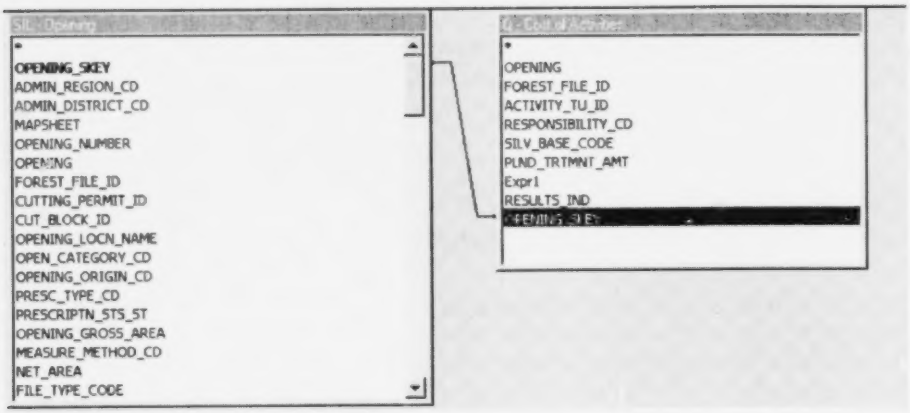
b) Run the query to test it, then save and close it.

c) Create a new query from the database window. When the *Show Table* dialog box is presented, add any table(s) required, then click the *Query* tab in the *View* area of the *Show Table* dialog box to list the available queries.


Show Table
Tables
Queries
Both
Add
Close
ISS_Q1 - Prescription Information
ISS_Q1 - Regen Delay Requirements Not Mk
ISS_Q1 - Regen Delay that Needs to be Set
ISS_Q1 - Stocking Standards
Q - PL Stems/WS Stems
Q - Treated Area by Fund/Base Code
Q - Trees by Year
Q-FS922 1st Largest Planting No and Spp
Q-FS922 1st Largest Planting Number
Q-FS922 2nd Largest Planting No and Spp
Q-FS922 2nd Largest Planting Number
Q-FS922 Forest Cover
Q-FS922 History Attributes

d) Select the query created in steps a) and b), then click *Add* to add the query to the Query design screen. Close the *Show Table* dialog box.

Create the relationship (link) between the query and the table in the query window. The most efficient join should connect the query and the table using a Surrogate key (such as *OPENING_SKEY*) which is common to both objects. Drag the surrogate key field from the table to the equivalent SKEY in the query. The screen below shows a typical join when the surrogate key field has been added to the original query.



e) Add the fields required in the query output to the query grid. Fields should be added in the order that they are to appear in the output, and in the correct order if a hierarchical sort is required. If you want to display the source table name for each field in the query grid, choose **View|Table Names Query**.

f) Add any criteria required to the criteria cell(s) in the appropriate column(s) in the grid.

g) Add any sort criteria to the sort cell(s) in the appropriate column(s) in the grid.

h) Run the query to make sure that the output is what you expect.

i) Use **File|Save As/Export...** to name and save your query.

Exercise - Creating a Query on a Query

In this exercise you will create a Query on a Query using the Silvicultural Systems table and the query called Q - Cost of Activities which you created in an earlier exercise.

1 From the *Query* panel of the database window choose *New.*

2 Choose *Design View* from the *New Query* window.

3 Double-click the *ISS – Prescription Silvicultural Systems* table from the *Add Table* dialog box.

4 Click the *Queries* tab in the *Show Table* dialog box, then select *Q- Cost of Activities* and add it to your query. Close the *Show Table* dialog box.

5 Notice that there is no join between the two objects in the query window. Since there are no surrogate keys in the original query you will need to create the linkage based on the *Opening* field. Drag the *Opening* field from the *ISS – Prescription Silvicultural Systems* table to the *Opening* field in the query to establish the link. (***Note**: It would have been better to include the OPENING_SKEY in the Q- Cost of Activities output to provide a more efficient link.*).

6 Choose **View|Table Names** to add the *Table Names* row to the query grid if it is not already displayed.

7 Double-click the asterisk in the *Q- Cost of Activities* box to add all of the output fields to the new query output automatically.

8 Add the following fields from the *ISS – Prescription Silvicultural Systems* table to the query grid
SILV_SYSTEM_ST
DISTURBANCE_DATE

9 Run the Query. You should get 26 records.

10 Save the query as: *Q - ATU Cost with Harvest Code/Disturbance Date.*

11 Close the query window.

12 Modify the *Q- Cost of Activities* query to include the *OPENING_SKEY,* then rebuild the *Q - ATU Cost with Harvest Code/Disturbance Date* using the *OPENING_SKEY* as the link. You should still get 26 records.

Summarizing Data Using a Query on Query

Sometimes it is easier to create a summary query which is based on an existing query. You create the first query, then create a summary query which is based on the first query. The following exercise demonstrates.

Exercise - Creating a Query on a Query

In this exercise you will summarize the Total Silviculture Accomplishments by Funding Source for 1993/1994.

1 Create a new query based on the *Activity Treatment Unit* table.

2 Add the following fields to the query output:
ACT_FUND_SRCE_CD
ATU_COMPLETION_DT
SILV_BASE_CODE
ACT_TRTMNT_AMT
RESULTS_IND

3 Add the following Criteria to the *ATU_COMPLETION_DT* criteria cell:
Between #1993/04/01# And #1994/03/31#

4 Add the following Criteria to the *RESULTS_IND* criteria cell:
="Y"

5 Run the query and check your output. You should get 181 records.

6 Save the query as *Q - Raw Accomplishments 93/94.*

7 Close the query.

8 Create a new query.

9 In the *Show Table* dialog box, select the *Queries* tab, then add the *Q - Raw Accomplishments 93/94* to the query. Close the *Show Table* dialog box.

10 Choose **View|Totals** (or click the *Totals* button on the toolbar).

11 Drag the *ACT_FUND_SRCE_CD, SILV_BASE_CODE,* and *ACT_TRTMNT_AMT* fields to the query grid. (The *ATU_COMPLETION_DT* field is not used in the output because you will not want to group the output by date).

12 Choose *Sum* as the operator in the *Totals* cell for the *ACT_TRTMNT_AMT.*

13 Run the query and study the output. You should get 21 records

14 Save the query as *Q - Accomplishments by Licensee 93/94*.

15 Close the query window.

Advanced Queries

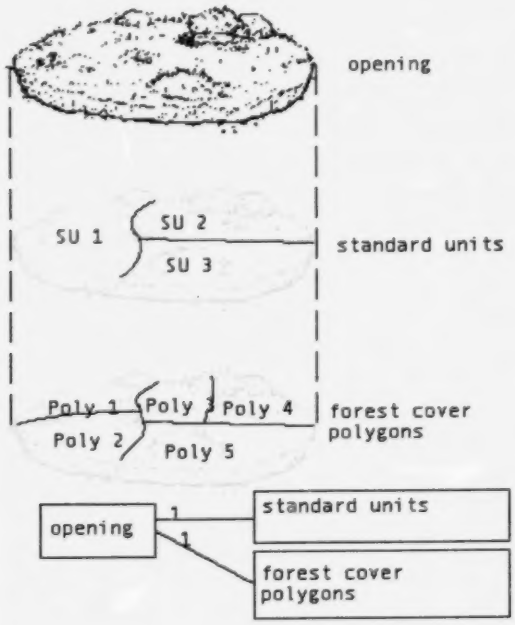
Understanding ISIS Relationships - Fundamental Rules

Certain business rules that have been defined in ISIS will impact on your reporting. At times, you are requested to provide queries/answers on subject areas where the business relationships are not well defined. For example, the desire to perform analysis on post-treatments impacts using the forest cover polygons label. Unless ATU's and polygons have SU link, it is very difficult to correlate that a certain treatment yielded certain forest cover results. Often some type of "built-logic" will have to be developed to "best-guess" this relationship such as pro-rating of areas.

The following is a list of all fundamental rules in ISIS. You should be familiar with these rules as they will greatly enhance your query building skills and the query output (e.g. one to many relationship commonly yield multiple row results).

Subject Areas	**ISIS Fundamental Rule**
Openings	• One opening can have one or more prescription (e.g. SP --> SMP ----> SP etc., but this is not well rationalized in ISIS yet). • One opening can have one or more tenures, although one tenure will be tagged as prime (typically the largest) for reporting and display purposes on the ISIS header. • One opening can have many standards units (SU). • One opening can have many (ECO) stratum units. • One opening can have many activity treatment units (ATU). • One opening can have many forest cover polygons. • One opening can have one or many historical land status records. Only the most recent land status area is currently available through SIA via the SIL-Opening Table.
Projects	• One project can contain many ATU's

Subject Areas	ISIS Fundamental Rule
ATU's	• One ATU can be a part of only one project. • One ATU can have links to multiple standard units. Note: If the SU links in ATUs are not used, it is very difficult to infer silviculture treatment to forest polygons (e.g. polygon has SU x-ref, ATU has SU x-ref). If neither is used, no relationships can be drawn.
Forest Cover	• One forest cover polygon can have links to only one SU. • Land status is derived from forest cover polygon information with date stamp of forest cover entry date. Note: Forest cover is not archived, only the most current forest cover is tracked.



Distinguishing Between Different Business Areas

With the merging of MLSIS information into ISIS, the requirement to understand how to build queries will be even more important.

Since a MLSIS record contains smaller amount of information than an ISIS record, there will be many blank records. Therefore, it is important to understand the differences between major licensees versus other records so that you know how to pull out the relevant business areas.

BUSINESS AREA	OPEN_ CATEGORY _CD	RESPONSI BILITY_ CD	DISTURBAN CE_ DATE	SILV_ FUND_ SRCE_ CD	FOREST_ FILE_ID	FILE_ TYPE_ CODE	APPROVAL_ DATE
SBFEP	FTSBF NDFS JHFS	F	>=1988-01-01	SBF		B20 B21 B22 B23 (A03)	>=1988-01-01 (not always filled)
ML	FTML VHML NDOML NDVML	L	>=1987-10-01	IA		A01 A02 A06	>-1987-10-01
Backlog SP (MOF)	BLFS	F	<1988-01-01	MO			>=1995-06-15 (not always filled)
Backlog SP (Licensee)	BLML	L	<1987-10-01	IO			>=1995-06-15 (not always filled)
Woodlots	FTWL	L	>=1987-01-01	IA	Like "W*"	A04	
Not Free Growing	P87	F or L	>=1988-01-01	VOI or VG		n/a	
PFG	FG	F or L	n/a	?		n/a	
SMP (MOF)	SMPFS	L	n/a	MO		n/a	>=1995-06-15 (not always filled)
SMP (Licensee)	SMPML	F	n/a	IO			>=1995-06-15 (not always filled)
Excluded	NREQ						
SP Exempted	SPEX						
SMP Exempted	SMPEX						

Common Data fields with different meanings

In SIA, there are many duplicated fields to make reporting easier by reducing the number of table joins. However, some fields look the same, but are in fact different in their meaning. When reporting on any of the following fields, you should pay special attention to whether you are using the right field to answer the right question.

Table	Field	Meaning
SIL-Opening	REGEN_DELAY_DATE	Derived from the latest Regen Delay Date of all SUs within the same SP.
SIL-Opening	REGEN_DATE	Date when the OPENING is declared Regen Achieved. Note that this field has only been implemented since 1998; therefore caution must be used when reporting on older areas for calculating average achieved regeneration delay.
ISS - Stocking Standard	REGEN_DELAY_YEARS	This is the regen delay years for the defined standards unit from the SP.
ISS - Stocking Standard	SU_REGEN_DATE	This is the achieved regeneration declaration date for the standards unit, entered via the Milestone Screen.
ISS-Stocking Standard	SU_REGEN_IND	If Date is entered in the Regen Achieved Date in the Milestone Summary Screen, the indicator will be "Y".
ISS-Prescription	REGEN_IND	If all SU's have an entered Regen Achieved Date, the indicator will be automatically switched to "Y".
SIL-Opening	EARLY_FG_DATE	Derived from the latest Early FG Date of all SUs within the same SP.
ISS - Stocking Standard	EARLY_FG_YEARS	This is the early free growing years for the defined standards unit from the SP.
SIL-Opening	LATE_FG_DATE	Derived from the latest Late FG Date of all SUs within the same SP.
ISS - Stocking Standard	LATE_FG_YEARS	This is the late free growing years for the defined standards unit from the SP.
SIL-Opening	FREE_GROW_DATE	Date when the OPENING is declared Free Growing Achieved.
ISS - Stocking Standard	SU_FREE_GROW_ DATE	This is the achieved free growing declaration date for the standards unit, entered via the Milestone Screen.
ISS-Forest Cover	FREE_GROW_IND	Declaration of the polygon if it is free growing. This is not linked to the Milestone Summary Screen declaration.
ISS-Stocking Standard	SU__FREE_GROW_IND	If the SU Free Growing Achieved date is entered in the Milestone Summary Screen, then the indicator will be switched to "Y".
ISS-Prescription	FREE_GROW_IND	If all SU's have an entered Free Grow Achieved Date, the indicator will be automatically switched to "Y" for the opening.
SIL-Opening	DISTURBANCE_DATE	This is based on the harvest start date for the entire opening.
ISS-Harvest Detail - Planned/Actual	ACT_HARV_START_DT	This is reported on Form "B" as the harvest start date for each denudation entry. There can be multiple harvest entries with different dates within one opening.
SIL-Opening	SILVIC_HANDOVER_ DT	This is based on the last harvest end date for the entire opening.
ISS-Harvest Detail - Planned/Actual	ACT_HARV_END_DT	This is reported from Form 'B' as the harvest end date for each denudation entry. There can multiple harvest entries with different dates within one opening.
ISS-Harvest Detail - Planned/Actual	RESERVE_TYPE_CD	This is reported on Form 'B' along with denudation areas.
ISS-Forest Cover	RESERVE_TYPE_CD	This is reported on Form 'C' by forest cover polygon/stratum.
SIL-Opening	OPENING_GROSS_ AREA	Opening Gross Area was the original ISIS Gross Area. This was not removed but should be the same as TAUP Area.
SIL-Prescription	TAUP_AREA	Total Area under the Prescription is based on information supplied on the SP. This is also the same as OPENING_GROSS_AREA.
TIM-Cutblock	CUR_GROSS_BLK_ AREA	Gross area is managed within FTAS.

Table	Field	Meaning
SIL-Opening	NET_AREA	Sum of all net area to be reforested (NAR) for all standards unit within the opening
ISS - Stocking Standard	TREATABLE_AREA	This is the Net Area to be Reforested for the defined standards unit
SIL-Opening	BGC_ZONE BGC_SUBZONE BGC_VARIENT BGC_PHASE BGC_SITE_SERIES BGC_SITE_TYPE ELEVATION_AVG SLOPE_AVG	Obtained from the largest ecology/SU link within the opening
ISS-Ecostrata	BGC_ZONE BGC_SUBZONE BGC_VARIENT BGC_PHASE BGC_SITE_SERIES BGC_SITE_TYPE ELEVATION_AVG SLOPE_AVG	Detailed ecology information entered by Stratum with cross-reference to standards unit

Breaking down the question to build the query

When someone asks you for information, you will need to match their question to the corresponding SIA tables. You will also need to ask yourself, what level of resolution is really required for the client, and how this information may be used. Often, it will take several steps to arrive at the desired answer. Take baby-steps when building your queries, to ensure your criteria are working properly.

Exercise Query on Queries

In the following exercise you will determine which backlog openings have NSR but no planned activities. This information could be used to start scheduling new activities.

Step 1. Identify backlog openings with NSR

1 Create a new query based on the *SIL-Opening* table

2 Add the following fields to the query grid:
OPENING_SKEY
DISTURBANCE_DATE
ADMIN_DISTRICT_CD
OPENING
OPEN_CATEGORY_CD
OPENING_GROSS_AREA
NET_AREA
NSR_TOT_AREA

3 Set the following criteria:

Field	Criteria	Reason
DISTURBANCE_DATE	< "1988/01/01"	This is the breakpoint of backlog blocks.
NSR_TOT_AREA	>0	This isolates only those backlog openings with remaining NSR.

4 Run the query and study the output. You should get 24 records

5 Save the query. (This query is saved for your reference in the Training Database as *Q – Step 1 -Backlog Opening with NSR*)

Step 2. Identify backlog openings with no planned activities.

To identify for no planned information, you need to flip the question around and look for those Openings with planned activities since this is easier to query.

1 Create a new query based on the *ISS-Activity Treatment Unit* table

2 Add the following fields to the query grid:
OPENING_SKEY
RESULTS_IND
DISTURBANCE_DATE

3 Set the following criteria

Field	Criteria	Reason
DISTURBANCE_DATE	< "1988/01/01"	This is the breakpoint of backlog blocks.
RESULTS_IND	N	This identifies those openings with planned ATUs
OPENING_SKEY		Surrogate Key is pulled to enable tying Step1 and Step 2 information together.

4 Run the query and study the output you should get 274 records.

5 Save the query. (This query is saved for your reference in the training database as *Q – Step 2 - Backlog Opening with Planned ATUs*)

Step 3 - Combining the Steps 1 and 2 using outer joins

If we call up Step 1 and Step 2 and complete a simple join between **OPENING_SKEY**, this would not give us the desired results. We would get all backlog blocks with NSR with planned activities. Since we want the reverse of this, we have be perform an **outer join**.

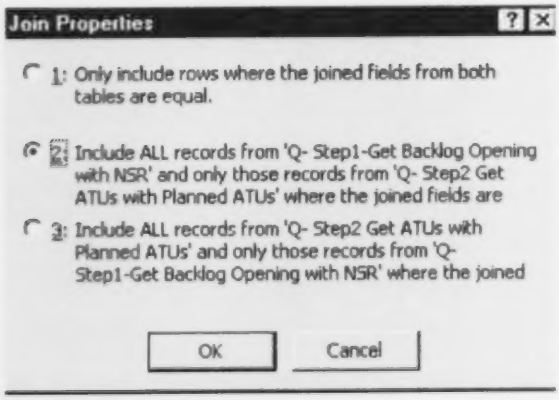
1 Create a new query using the two queries created in steps 1 and 2. (*Q – Step 1 -Backlog Opening with NSR* and *Q – Step 2 - ATUs with Planned ATUs in the training database).*

2 Join the two queries via the OPENING_SKEY

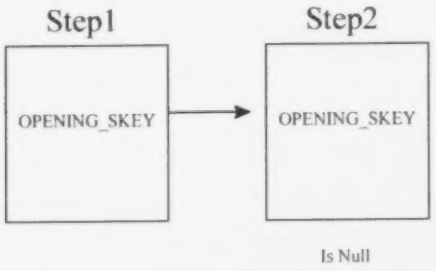
3 Add the following fields in the following order from the query created in step 1 to the query

ADMIN_DISTRICT_CD
OPENING
OPEN_CATEGORY_CD
DISTURBANCE_DATE
OPENING_GROSS_AREA
NET_AREA
NSR_TOT_AREA

4 **Double click on the join** on the **OPENING_SKEY** between the two tables, to select a join from the Join Properties box, whereby the arrow points from NSR Step 1 to Planned Activities Step 2. This will show all cases of NSR with and without planned activities.


Join Properties
1: Only include rows where the joined fields from both tables are equal.
2: Include ALL records from 'Q- Step1-Get Backlog Opening with NSR' and only those records from 'Q- Step2 Get ATUs with Planned ATUs' where the joined fields are
3: Include ALL records from 'Q- Step2 Get ATUs with Planned ATUs' and only those records from 'Q- Step1-Get Backlog Opening with NSR' where the joined
OK
Cancel

5 However, the records with no planned activities will have blanks associated with the Step 2 field columns. To isolate for those records without planned activities, add the RESULTS_IND from the query created in step 2 to the query and set **IS NULL** criteria to only isolate the blanks. This will list of openings that will be of interest with no planned activities.



6 Run the query . You should get 4 records.

7 Save the Query. (This query is saved for your reference in the training database as *Q – Step 3 -Backlog Opening with NSR)*

Exercise Aggregate Queries

Aggregate (Total) Queries are used to provide bulk totals, averages and counts of detailed query results. They are often done as the last step after confirming detailed results. In this exercise you will calculate the total trees and hectares planted for the 1998 calendar year.

1 Create a new query based on *the ISS – Planting Details – Planned/ Actual* Table

2 Add the following fields to the query grid
ACT_PLANTED_NO
ACT_TRTMNT_AMT
RESULTS_IND
ATU_COMPLETION_DT

3 Set the *Total* cell for *Sum* for the ACT_PLANTED_NO field and the ACT_TRTMNT_AMT field. Leave the *Total* cell set to *Group by* for the RESULTS_IND field. Set the criteria for the RESULTS_IND field to ="Y"

4 Set the *Total* cell to *Where* for the ATU_COMPLETION_DT field and set the criteria to Between #1998/01/01# And #1998/12/31# NOTE: make sure that the "show" field is checked.

5 Run the query.

6 What is the total number of trees planted in the 1998 Calendar year?_____

7 What are the total hectares planted in the 1998 Calendar year?
_____

8 Save the Query. (For your reference this query has been saved in the training database as *Q - Aggregate Query - Total Trees and Areas Planted)*

Understanding the evolution of data requirements and the impact on reporting

The long lifespan of silviculture prescriptions and the associated tracking of silviculture responsibilities and activities present a challenge in the data management world. Stands are required to be tracked up to 15 years to ensure free growing obligations are met. Furthermore, some of these areas have stand management prescriptions and with subsequent activities.

The long time frame and the continued evolution of Forest Practices Code (FPC) place high demands on data management. As such, there are shifts in data requirements depending on the timeframe we are inquiring about. For example, the requirement for maximum density occurred in 1995. Prior to this date, no maximum density values were entered, though it appears in the current ISIS screens as today's standard. Another example is that site index has replaced site class although there were no direct conversion from site class to site index. Site index is gradually getting filled over time because it is required of new survey procedures.

A change in data requirements does not necessarily mean that all historical information is converted to the current standards. Sometimes, data conversion is done to move data to new standards, if this is possible. But for many older blocks, a more typical scenario is to have the new standard field left blank since the standard for entry was never required. This poses challenges for reporting, as you will have to deal with these data gaps. When running reports, you must have some understanding of why these data gaps occur (e.g. a problem with data entry versus a problem due to shifts in data capture standards). Qualifying these differences would be of interest to your clients.

The following table shows some significant milestone events that trigger subsequent events in the data world. This will assist your appreciation of what must be considered when running reports.

Event	Date	Impacts
Silviculture Treatment Reporting Policy	Areas disturbed before October 1, 1987 under major licences or woodlot licences	All records prior to this date will have little or no-stocking standard requirement, unless it is picked up as a backlog silviculture prescription Old areas logged by licensee between January 1, 1982 and October 1, 1987 were considered to be the "Industry Outstanding" obligations. Currently, they are identified as the "backlog areas".
Requirement for ML and woodlot holders to establish a free growing stand	Areas disturbed from October 1, 1987 up to June 15, 1995 under major licences and woodlot licences	Requirement for an approved PHSP prior to harvest commencement. The reporting of ecology information, stocking standards, harvesting, silviculture and forest cover information was entered via MLSIS and subsequently converted to ISIS.
Requirement for SBFEP to establish a free growing stand	SBFEP logged area logged after January 1, 1988	All records prior to this date will have little or no stocking standard requirement, unless it is picked up as a backlog silviculture prescription. Old areas not logged by the licensees prior to January 1, 1988 were considered to be the old "Ministry Outstanding" obligations. Currently, these are identified as the "backlog areas".
Requirement for SBFEP to establish a free growing stand	Areas disturbed from October 1, 1987 up to June 15, 1995 under SBFEP	Requirement for an approved PHSP prior to harvest commencement. The reporting of ecology information, stocking standards, harvesting, silviculture and forest cover information was entered via History Records and subsequently converted to ISIS.
Requirement to establishing a free growing stand	Areas disturbed post June 15, 1995	The data requirements for SP has shifted over time; as such there will be differences in the information found in the SP portion ISIS.
SMP requirement for stand enhancement treatments on young stands.	Areas with stand enhancement activities (e.g. incremental spacing, fertilization, and/or pruning) after June 15, 1995	The entry of stand management prescriptions into ISIS has not been standardized. As such, the labelling opening category and reporting of activities exist; however, all other information such as stocking standards are intermittent.

Business Applications of SIA

Emerging Uses of SIA to meet other business areas needs

The information collected in ISIS is useful for serving other processes such as developing Resource Management Plans, assisting in the Timber Supply Review through calculating actual achieved regeneration and data reconciliation. Different areas are starting to develop canned queries, which may facilitate faster completion of this work within the district.

Reconciling FTAS and ISIS Information

Information Access infrastructure provides easy cross-referencing of information between FTAS and ISIS. FTAS's access database called Timber Information Access (TIA) is extracted biweekly and sent to the same district LAN location as SIA. The CUTBLOCK table is a key reference table for silviculture records containing key identifiers, areas, dates and block status useful for the silviculture business. Some periodic checks that can be performed are as follows:

- identify cutblocks logged, but not yet in ISIS; theoretically, all completed harvested blocks should have a corresponding record in ISIS, unless they are SP Exempt.

- reconcile areas and dates between the two systems. Future streamlining and work on reducing data duplication under INCOSADA can use SIA/TIA tools to assist in reconciling information between the two systems.

Currently sets of queries are available to locate ISIS records in SIA by tying to TIA dataset. Simple Attach Link Tables will enable queries to access different datasets.

Canned SIA queries to assist in locating data gaps are as follows (see *Appendix A* for full description of queries):

- FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, Blk, CP
- FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, Blk
- FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, CP
- FOR_Q1 – FTAS SBFEP Blks Xref to Opening (Attach Cutblock Table)
- FOR_Q1 – Possible SBFEP Blks Not in FTAS (Attach Cutblock Table)

Exception Reports or Locating Missing Information

A common use of SIA is to locate missing information or exception reporting. To successfully build these queries, you will have to understand how to breakdown the question, and perhaps ask the same question but in a different way to formulate the proper querying steps.

As in the NSR example, the challenge was to locate those NSR with no planned activities. To accomplish this, we re-phrase the question to locate NSR with planned activities. We then reversed the results via the use of an outer join to find the missing information.

Another example is the case of SBFEP data gap identification. We need to identify all those SBFEP openings in ISIS. Then find all those SBFEP blocks in FTAS. Then when we combine the SBFEP FTAS data set against the ISIS SBFEP data set, we look for gaps between the two. These gaps are of interest, since we need to ensure that these blocks, if not SP exempt, are entered into ISIS for inspections and compliance purposes. Again, outer joins are used to assist in identifying for data gaps.

Canned SIA queries to assist in locating missing information are as follows (see *Appendix A* for full description of queries):

- Exception-ISIS Openings with Harv St Dt & ATU & No Forest Cover
- Exception-ISIS Openings with Harv St Dt & Forest Cover and No ATU
- Exception-ISIS Openings with Harvest Start Dt & No Forest Cover
- Exception-ISIS Openings with SUs and No Regen Delay
- Exception-ISIS Openings with SUs and No Early FG Yrs
- Exception-ISIS Openings with SUs and No Late FG Yrs

Exercise – Exception Reporting

In this exercise you will determine ISIS Openings with no SU's

1 Create a new Query based on *SIL-Opening* table and the *ISS- Stocking Standard* table

2 Add the following fields from the *SIL-Opening* table
ADMIN_DISTRICT_CD
OPENING
OPEN_CATEGORY_CD
OPENING_GROSS_AREA
DISTURBANCE_DATE

3 Add the STANDARDS_UNIT_ID fields from the *ISS- Stocking Standard* table

4 Set the criteria for OPEN_CATEGORY_CD field to - Not In ("P87","NREQ","CONT","FG") and the criteria for DISTURBANCE_DATE to <>#9999/12/31#

5 Set the criteria for the STANDARDS_UNIT_ID field to Is Null

6 Double click on the join between the two tables. Choose an "outer Join" (#2 in the Join Properties box)

7 Run the query. You should get 3 records

8 Save your Query. (For your reference this query is already saved in the training database as :*Q - Exception-ISIS Openings with No SU's)*

Routine District Monitoring and Compliance Reporting

It is a fairly simple step to run your own monitoring and inspection reports for those areas that are due for regeneration or free growing and indicate if they have been achieved. This style of routine reporting can assist districts in planning their workloads, and working with their licensees to ensure that they are aware of which blocks/standards units are coming due for the upcoming year.

Canned SIA queries to assist in monitoring and compliance reporting are as follows (see *Appendix A* for full description of queries):

- ML-Compliance Reporting – List of SU with Regen Due
- ML-Compliance Reporting –List of SU with Early FG Due
- ML-Compliance Reporting-List of SU with Late FG Due

Optional Exercise – Compliance Reporting

Determine which ML's have SU's with Late FG due

Hints:

Use three tables for this query *ISS-Stocking Standard* table, *SIL – Opening* and *ISS Prescription table*
Sort in Ascending order by FOREST_FILE_ID

Set the criteria for OPEN_CATEGORY_CD to "FTML"

Use the following formula to calculate the late FG Date due -
Late FG Due Yr: Year([SIL-Opening]![DISTURBANCE_DATE])+[LATE_FG_OFFSET]
Set Parameters for the Late FG Due Yr to < 2002

Save your Query . (This query has been saved for your reference in the training database as *ML-Compliance Reporting - List of SU with Late FG Due.)*

FS922

For those districts that have not yet been converted to VEGCAP, a FS922.mdb file has been developed. This database contains associated queries and modules to generate a report that looks similar to the FS922 to assist in the inventory update process. This saves the time required to transcribe forest cover information, by hand, into the FS922.

A database called FS922.mdb file and an associated Word file, describing the contents of the mdb file, is available on the IMG's IAC/SIA website.

Resource Management Plans-Guidelines for the Sustainable Harvest Component

Resource Management Plans are submitted annually by districts to assist in identifying management objectives and priorities for investment opportunities.

SIA can assist in summarizing completed silviculture activities and future scheduled activities. A database called RMP-Sustainable Harvest Component.mdb file and an associated Word file, describing the contents of the mdb file, is available on the IMG's IAC/SIA website. This database contains standarized queries to support the generation of the Resource Management Plans.

TSR-Regen Assumptions for managed stands

Timber Supply Branch requires district submission of regeneration statistics for the timber supply model. Some of this information can be derived via SIA. Forest Practice Branch is currently working to provide Timber Supply Branch with SIA access. This will the allow the Timber Supply Analyst to have a better understanding of silviculture information, enable more standardization of data extract/outputs and facilitate dialogue between districts and the timber supply analystson how to best obtain information for conducting timber supply analysis.

A database called TSR-Regen Assumptions.mdb file has been developed to address some of the regeneration assumptions of the TSR data package. An associated Word file describes the contents of the mdb file and is available on the IMG's IAC/SIA website.

Optional Exercise – Aggregate Query

Calculate the average density of trees planted by calendar year

Hints:

Break the query into two steps

In step 1:

- determine the average density by ATU (using *the ISS- Planting Details Planned/Actual* table)
- Use the Year function to convert the ATU completion date to a year- PL Year: Year([ATU_COMPLETION_DT]) >0
- Use the following formula to determine the density - Density: [ACT_PLANTED_NO]/[ACT_TRTMNT_AMT] at fixed 0 decimal places
- You should get 463 records when you run this query
- Save your query for use in Step 2. (For your reference this query is saved in the training database as *Q-Aggregate Query - Step1 Avg Density by ATU.*)

In step 2:

- Create a new query based on the query established in Step 1
- Group by the PL Year
- Average by Density
- Format the density field to have fixed zero decimal places
- Run the query – you should get 24 records
- Save your query. (For your reference this query is saved in the training database as *Q- Aggregate Query - Step2 Avg density by Year)*

What is the average density for trees planted in 1978, 1988 and 1998?_____

Module Building

Modules are mini-programs you can build to assist in doing common repeated functions or complex calculations. Modules follow very strict conventions and rules. Modules have the following components:

- ***Function name declaration*** – this provides the name of the function. This is the trigger name used in the expression in your query. The function name should not be the same as the Module name.
- ***Function output datatype*** – this specifies the kind of datatype returned as output (i.e. date, integer, single (decimal), double (decimal) etc.)
- ***Function statement*** – this details the calculation to be preformed and what will be returned as output
- ***End Function Statement*** – this tells the module to finish

Note that once a function has been created it can be accessed as a custom function via the Expression Builder.

Example of a simple function:

Function FiscalYr(anyDate) converts yyyy/mm/dd field into fiscal year

FiscalYr=IIf((Month(anyDate)>=4),Year(anyDate)+1, Year(anyDate))

End Function

See the module *SIL - FiscalYr* to view the function and the query *Module - Function FiscalYr* in the Training Database for an example of how this function is used

Complex functions

Sometimes you need to evaluate for multiple variables and multiple scenario's. Complex functions allow for this by the use of IF..THEN and SELECT..CASE statements which are structured in the following manner:

```
IF variable meets criteria THEN
END IF
SELECT variable meets criteria
        CASE equals value or range of values
                If condition is true, then variable is assigned value
        CASE equals value or range of values
                If condition is true, then variable is assigned value
        Etc. use as many CASE statements as required
END SELECT
```

Example of a complex function:

The function *GreenupSIHt* calculates the annual height growth based on two variables, tree species and site index. The function is set up to pass two variables which are identified as:

- ***CurrentSpp*** - which has been set to a data type of string (text)
- ***SI*** - which has been set to a data type of integer

(Note that you can assign as many variables as needed in your declaration statement, as long as you define them with the associated data type within the round parenthesis.)

The function *GreenupSIHt* is then set to pass double data type (represents a double precision floating point up to 8 bytes in storage size) into the output.

An IF statement is used to identify what species type occurs, then a SELECT CASE statement is used to specify the range of site index for the given species type and what annual height growth will get assigned to a given species and site index condition.

```
Function GreenupSIHt(CurrentSpp As String, SI As Integer) As Double
    If CurrentSpp ="Fdi" then
    Select Case SI
        Case 1 to 10
        GreenupSIHt =0.1
        Case 11 to 20
        GreenupSIHt =0.2
        Case 21 to 25
        GreenupSIHt =0.3
        Case 26 to 30
        GreenupSIHt =0.4
    End Select
    End If
    End Function
```

There are a number of different module examples available in the training database and attached in the *Appendix B* of this guide for your reference.

Module and Function Tip:

Modules and functions are not easy to build and require patience and lots of debugging. The specifics of building modules/functions is beyond the scope of this SIA Training Course.

If you need to develop modules due to complex analysis requirements, please review the modules/functions that are provided within the training database and the bi-weekly extracts. If at all possible, try to use these as "skeletons" to save time in building any modules/functions from scratch. Make adjustments to function name, variable, data type and conditions of functions as required. Descriptions of ready built modules which are available within SIA are shown in *Appendix B*.

Managing Queries

After a short period of working with SIA, you will have established a growing list of queries which will need to be managed.

It is suggested that you develop a consistent naming convention for your queries. For example, in this course all queries have been named using "Q - " as the first part of the query name. This helps to differentiate between queries and tables. Use a designation such as "Q-MTQ"

It is also suggested that you develop some kind of cataloging system to group similar queries together. For example, queries based on ISIS data could be named "ISS - Q1" followed by a description of the query content.

Any queries which are no longer used should be deleted from the database by selecting each unused query in the database window and pressing <Delete>.

Infrequently used queries can be stored in a separate database and imported when needed. Once the query has been used it can then deleted from the working database, and re-imported the next time it is needed. Importing queries is covered in the "Setting up your PC" section of this guide.

Another suggestion is to set-up separate .mdb files by themes. For example, resource management plans, business planning, sbfep all can have their own unique .mdb files so that all queries are bundled together by theme. There is relatively little storage space overhead for creating new databases since we are link to the LAN tables. This might provide an easier form of organization versus trying to manage one database with a massive amount of queries through different naming convention. Note that if you create new .mdb files by theme, you will have to create links to your LAN SIA file. However, this takes very little set-up time for the potential gains in organizing your query information.

Reports

Introduction

Whenever you create a query, the output from the query is displayed in datasheet view in a columnar layout.

You can either preview or print the basic layout of query output directly from the datasheet window by choosing **File|Print Preview** or **File|Print** (or by clicking the appropriate buttons on the toolbar).

The printout from a query is fast, but does not provide any sophisticated formatting capabilities such as headers and footers. You can change the font for the entire printout, and manipulate column widths and row heights, however for more sophisticated hard copy you will need to generate a report.

Creating Reports

Access provides a sophisticated report generator which allows you to completely customize your hard copy output.

It is not the purpose of this course to provide a detailed explanation of all of the features of the report generator, however Access provides a set of "Report Wizards" which will assist you in the creation of your reports.

The following exercise will illustrate the basics of using a Report Wizard.

Exercise - Creating a Report

In the following exercise you will create a report based on the query Q - Treated Area by Fund/Base Code *which you created in an earlier exercise.*

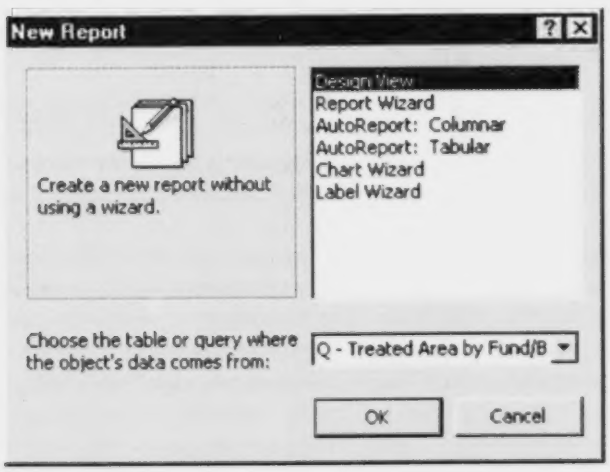
1 Open the query entitled *Q - Treated Area by Fund/Base Code* in design view.

2 Run the query.

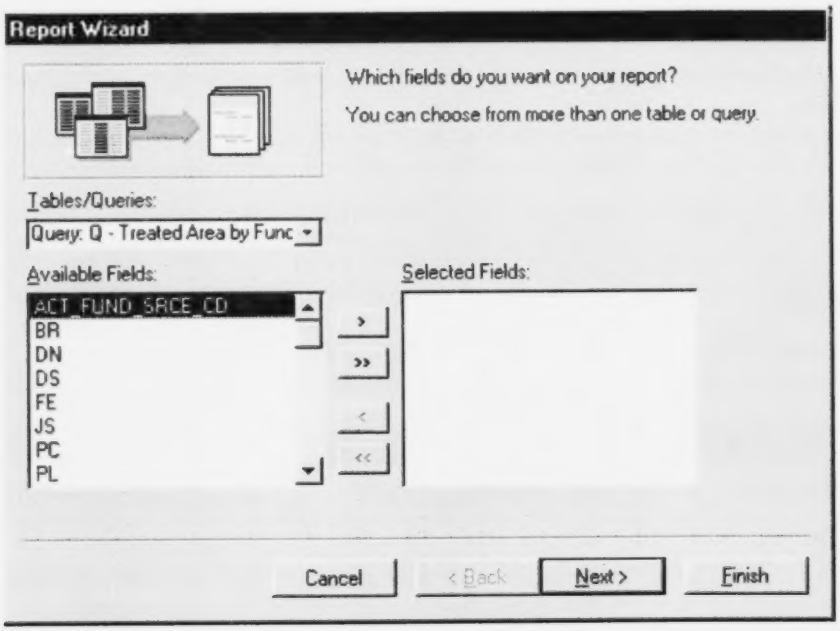
3 Choose **File|Print Preview** to view the printout in its current format.

4 Click the *Close* button on the toolbar .

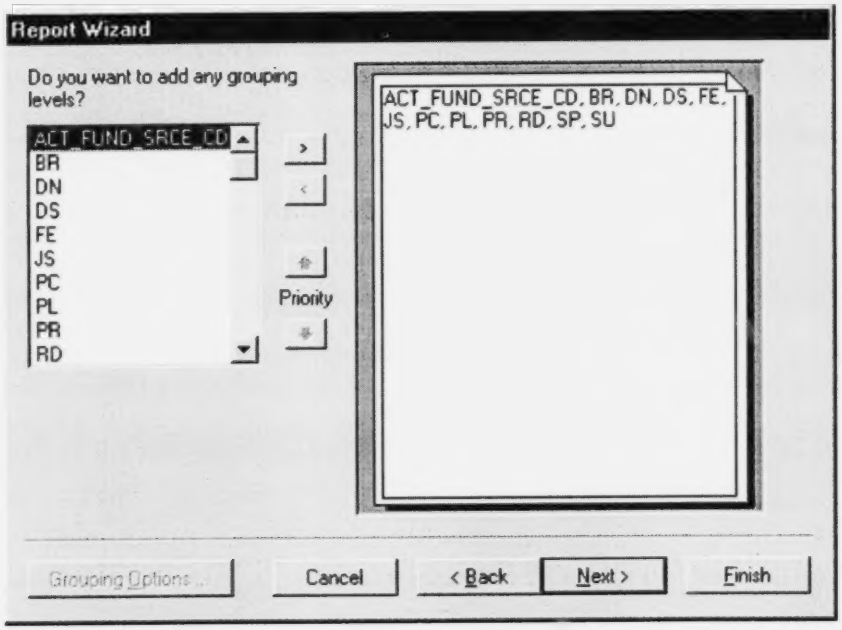
5 Click the *New Report* button on the toolbar. Access will display the *New Report* dialog box shown below:



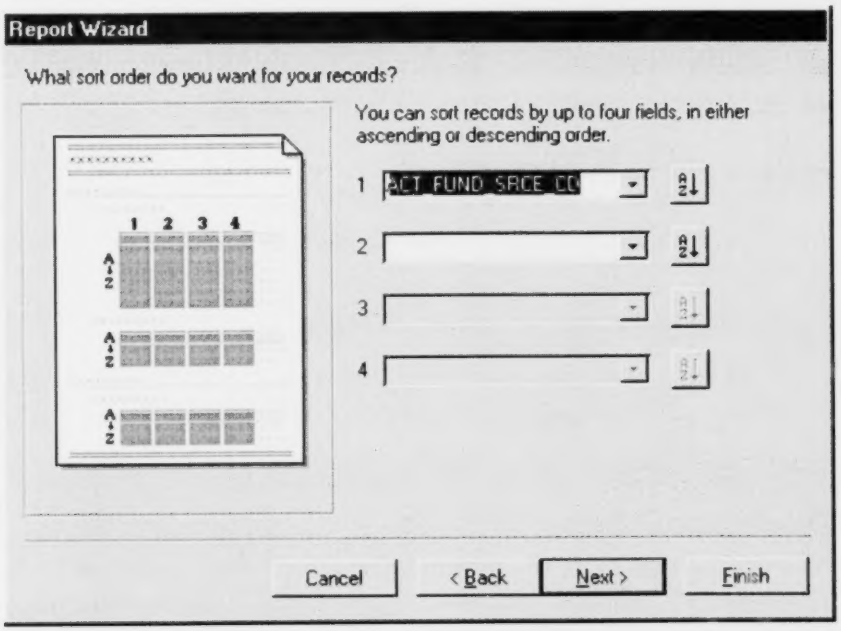
6 Select *Report Wizard*. Access will display the *Report Wizard* dialog box shown below. Click the button with the chevron (>>) on it to add all fields to the report output, then click *Next*.



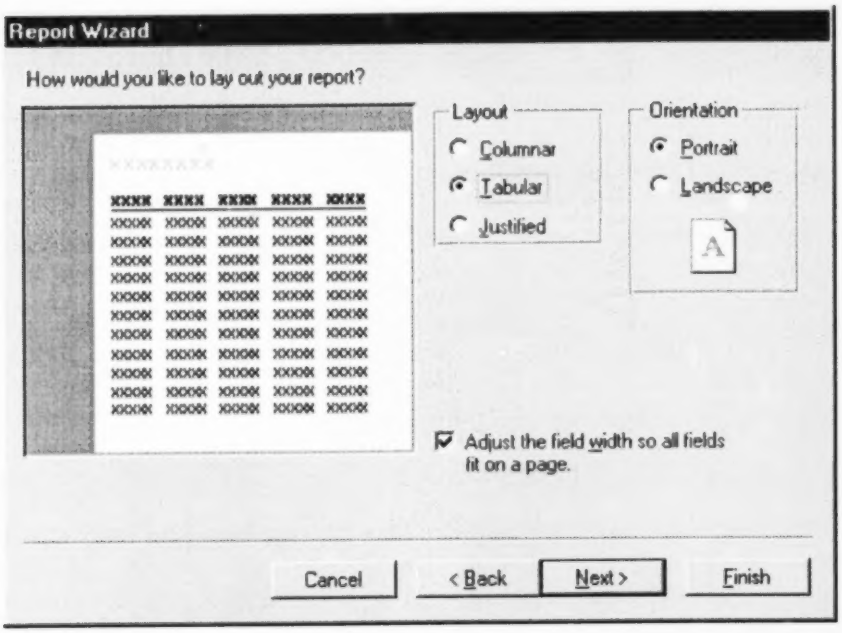
7 Access will display a grouping levels dialog box, to establish grouping levels for the report. Do-not select a grouping level for this report. Click *Next*.



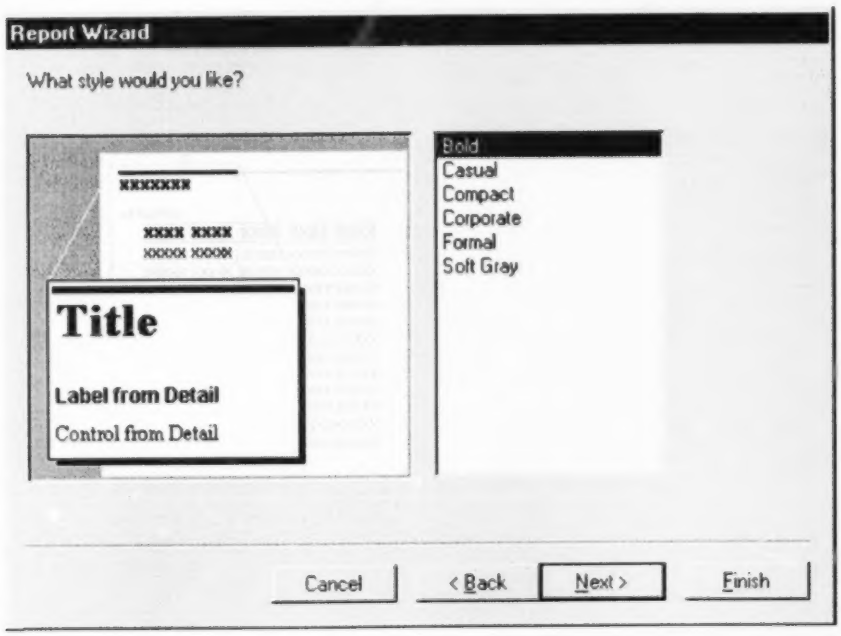
8 Select *ACT_FUND_SRCE_CD* field in the first field to establish the sort order for the report, then click *Next*.



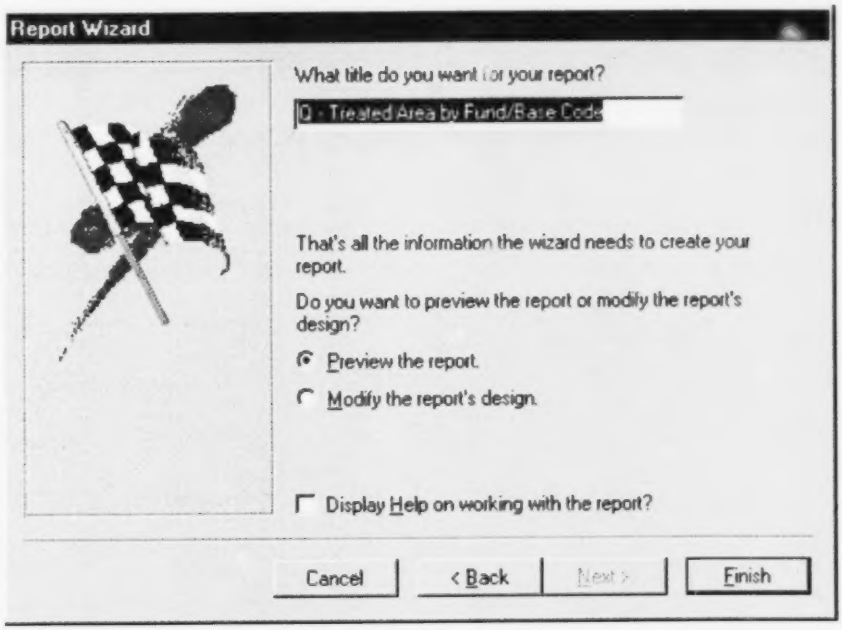
9 Select *Tabular* and *Portrait* for your report layout options. Also Select *Adjust the field width so all fields fit on a page* then click *Next*.



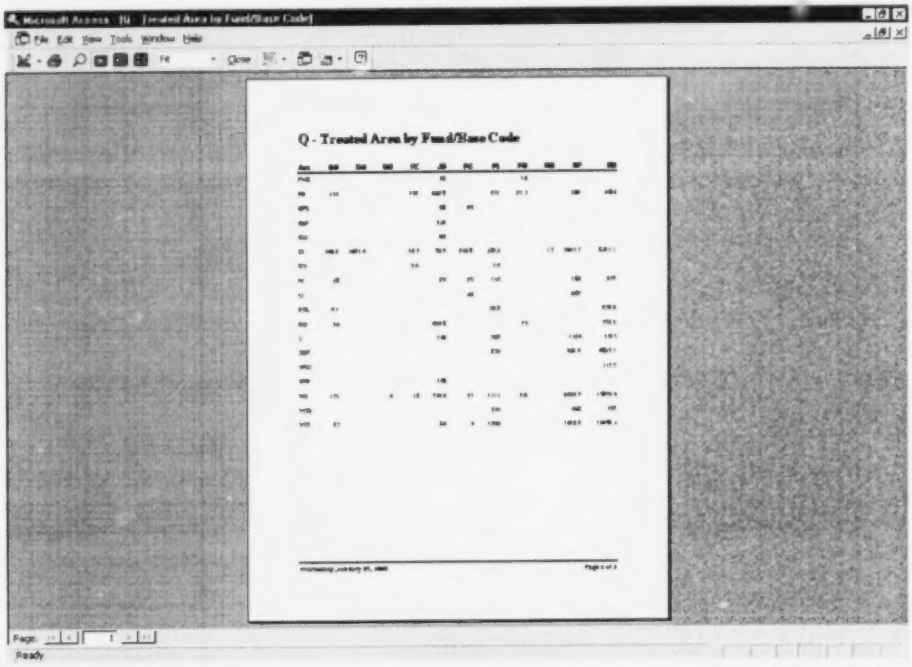
10 The following screen allows you to select a style for your report - select *Bold* and then click *Next*.



11 Select *Preview the Report* then click *Finish.*


Report Wizard
What title do you want for your report?
D - Treated Area by Fund/Base Code
That's all the information the wizard needs to create your report.
Do you want to preview the report or modify the report's design?
Preview the report.
Modify the report's design.
Display Help on working with the report?
Cancel
< Back
Next >
Finish

12 Be patient while Access creates your report. The report will be displayed in the *Print Preview* window. Maximize this window so that you can view the report. Use the magnifier to zoom in and out to preview sections of the report. Your screen should be similar to the screen shown below:



13 Choose **File | Save As/Export...** and name the report *R - Treated Area by Fund and Base Code.*

14 Close the report and the query.

15 Select the *Report* object button in the database window. Notice that your report is listed in the database window.

16 Double-click the report. Notice that Access runs the report without you having to open the query - the query that the report is based on is saved as part of the report parameters.

17 Close the Report.

On Your Own

Introduction

The following exercise will let you practice what you have learned in this course by allowing you to build and modify several queries without any step by step instructions. Create the queries, test them, then save them using an appropriate query name.

Exercise - Independent Practise

1 Create a new query which will list all openings logged before 1982.

2 Run the query.

3 Modify the query to prompt you to enter the disturbance type when the query is run.

4 Create a new query which will list all activities completed in fiscal year 1994 sorted by base code.

5 Run the query.

6 Update the query to provide a total for the treatment area by base code.

7 Create a new query which will list all 1993 survey project information with completed ATU's.

8 Run the query.

9 Create a new query which will list all planting ATU's and planting stock allotments. Modify the table join so that no rows are dropped from the output.

10 Run the query.

Troubleshooting, Tips and Support

Troubleshooting

Trouble shooting problems with SIA can be a difficult task since there are many areas where difficulties can occur.

The following list outlines some of the more common problems that you may experience when developing queries.

Syntax errors: Access can be very unforgiving when you are entering criteria into the query grid. If the results of your query are not what were expected, check the syntax of your criteria carefully. Syntax rules are discussed on page 52 of this manual.

To create successful queries, you must be very familiar with the data structures within SIA. Printing the data dictionary provides a useful hard copy resource for checking which tables contain the necessary fields, and also reminds you of the *field type* which is crucial to entering criteria using the proper syntax.

A useful strategy for solving syntax problems is to remove all criteria from the query, then begin by adding one criteria at a time. After entering each criteria, run the query and examine the output. This "divide and conquer" strategy will help you to isolate the specific criterion which is creating the problem.

Fields pulled from the wrong source table: In multi-table queries, the fields you choose can have significant impact on the query output. Select **View|Table Names** to display the source table names in your query grid. Determine which table contains most of the necessary fields to obtain the required output, and pull as many fields as possible from this main source table.

Incorrect Linkages between Tables: Link tables using the SKEY fields whenever possible. Make sure you link to the right SKEY. Remember that MS Access will allow you to link to anything. Garbage links will give you garbage results.

Too Many Tables: Delete any unnecessary tables from your query. Extra tables can produce unexpected results, slow down the execution of your query, and can create clutter and confusion in the query window.

Don't aribitarily change table, query or field names: When you build queries and reports, it is based on names used at the time of query construction. If you decide to change table, query or field names down the road, you will have negative impacts on all existing queries you have built already. Be very careful about changing names since you might have to go back and rebuild or fix your queries so they work based on the new name changes. Rule of thumb, think long and hard about naming your queries. Once you named them, don't change them, unless you really have to.

Functions and Expressions don't work: Remember that functions and expressions are to be used in the field row of the Query By Example Grid. Don't try to create functions and expressions within the criteria row.

Tips and Tricks

Zoom	**<Shift> F2**	To view expressions and criteria
Use Expression Builder		Easier to pick and choose fields than to type it in by hand. It is easier to save query first. Then all fields are easy to pick from the Expression Builder.
Ensure Queries have SKEYS		If you are going to use the query for subsequent queries, always drag down the appropriate SKEYS that you will need for future links.
Implied AND vs. OR		When you enter criteria across the query grid, there is an implied "and" relationship being developed, if the criterion is entered within the same row. Versus when you enter criterion vertically downwards, there is an implied "or" relationship.
Check your query results!		Always check your query results against ISIS data to ensure that you got the right results.
Integrated Data dictionary		**IMG has a website containing an integrated data dictionary (IDD).** If you go into the table design view, you can see field definitions. But the IDD might be of use if you want to do quick searches on table and field names if you are trying to look for specific information:

IDD website: http:\wwwinternal.for.gov.bc.ca/isb/datadmin/idd.htm

Support

If you require support, please contact:
Application Support Center (APPHELP) 387-8888
It is always best to export your query(s) you are having problems within into a new .mdb file (shell) and send this file to APPHELP along with a description of your problem. This makes debugging the query a lot more easier for the support center staff.
If you cannot locate the database on your LAN server, contact your district systems person.

Appendix A

List of Canned SIA Queries & description

FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, Blk, CP

Uses SIL-Opening and TIA's Cutblock table. Will provide output for those ISIS openings which have an assigned licence no., cutting permit, and cutblock number which do not exist in FTAS.

FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, Blk

Uses SIL-Opening and TIA's Cutblock table. Will provide output for those ISIS openings which have an assigned licence no. and cutblock number only (no CP assigned) which do not exist in FTAS.

FOR_Q1 – SBFEP Blks Possible Not in FTAS – Links Licence, CP

Uses SIL-Opening and TIA's Cutblock table. Will provide output for those ISIS openings which have an assigned licence no. and cutting permit number only (no cutblock assigned) which do not exist in FTAS.

FOR_Q1 – FTAS SBFEP Blks Xref to Opening (Attach Cutblock Table)

Uses SIL-Opening and TIA's Cutblock table. Will identify those potentially SBFEP File Type Codes in FTAS (In ("A03","A20","A21","A22","A23","B05","B20","B21","B22","B23") and identifies associated ISIS assuming that there is assigned licence no, cutting permit and cutblock numbers. This provides a potential total SBFEP block population in FTAS and in ISIS.

FOR_Q1 – Possible SBFEP Blks Not in FTAS (Attach Cutblock Table)

Uses SIL-Opening and TIA's Cutblock table. Will identify those potentially SBFEP openings in ISIS which do not exist in FTAS assuming that that there is assigned licence no, cutting permit and cutblock numbers.

ML-Compliance Reporting – List of SU with Regen Due

Uses SIL-Opening, ISS-Prescription, and ISS-Stocking Standard. Will pull out FTML (major licencee) opening category based on a regen due year specified by the user. A parameter query has been set up to calculate Regen Due Year based on ((Year([DISTURBANCE_DATE])+[REGEN_DELAY_YEARS). This report will provide a summary of regeneration status by standards unit with a display of the SU milestone summary and NSR summary for the opening.

ML-Compliance Reporting –List of SU with Early FG Due

Uses SIL-Opening, ISS-Prescription, and ISS-Stocking Standard. Will pull out FTML (major licencee) opening category based on a early FG due year specified by the user. A parameter query has been set up to calculate Early FG Due Year based on ((Year([DISTURBANCE_DATE])+[EARLY_FG_YEARS). This report will

provide a summary of free growing status by standards unit with a display of the SU milestone summary and free growing area summary from land status for the opening.

ML-Compliance Reporting –List of SU with Late FG Due

Uses SIL-Opening, ISS-Prescription, and ISS-Stocking Standard. Will pull out FTML (major licencee) opening category based on a late FG due year specified by the user. A parameter query has been set up to calculate Late FG Due Year based on ([Year([DISTURBANCE_DATE])+[LATE_FG_YEARS). This report will provide a summary of free growing status by standards unit with a display of the SU milestone summary and free growing area summary from land status for the opening.

Q-FS922 1st Largest Planting Number

Uses ISS-Planting Stock Allotment. Takes the largest planting number within the ATU.

Q-FS922 1st Largest Planting No and Spp

Uses ISS-Planting Stock Allotment and Q-FS922 1st Largest Planting Number. Relates the largest planting number within the ATU to obtain species information. This is used to show the dominant species planted within the ATU.

Q-FS922 2nd Largest Planting Number

Uses ISS-Planting Stock Allotment. Takes the second largest planting number within the ATU.

Q-FS922 2nd Largest Planting No and Spp

Uses ISS-Planting Stock Allotment and Q-FS922 2nd Largest Planting Number. Relates the 2nd largest planting number within the ATU to obtain species information. This is used to show the secondary dominant species planted within the ATU.

Q-FS922 Forest Cover

Uses SIL-Opening and ISS-Forest Cover. Pulls all forest cover information displayed in the FS922 into the query for display in the FS922 report. Note that key fields have been set-up with parameter queries to allow for user entry. These fields are as follows:

Mapsheet {format for entry must be blank leading if 2 digit mapsheet number (e.g. _92K02500)}.

Opening number ({format for entry must be blank leading of not four characters (e.g. _ _ 15)}.

Forest File Id

Cutting Permit

Timber Mark

Cutblock ID

Q-FS922 History Attributes

Uses SIL-Activity Treatment Unit, Q-FS922 1st Largest Planting No and Spp and Q-FS922 2nd Largest Planting No and Spp. Summarises harvesting and silviculture activity treatments within the opening with disturbance year, associated areas and planted species. This query is used to show any completed harvesting or silviculture activities in a format similar to the FS922 History attribute section. Note that this query makes use of module FS922 ConvertActivityCodes to translate ISIS activity codes into codes used in the FS922.

Exception-ISIS Openings with Harv St Dt & ATU and No Forest Cover

Uses SIL-Opening, ISS-Activity Treatment Unit, and ISS-Forest Cover. Takes any opening with a disturbance date and opening with results recorded against it, but no forest cover polygons. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with Harv St Dt & Forest Cover & No ATU

Uses SIL-Opening, ISS-Activity Treatment Unit, and ISS-Forest Cover. Takes any opening with a disturbance date and opening with forest cover recorded against it, but no ATU with results. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with Harv St Dt & No Forest Cover

Uses SIL-Opening and ISS-Forest Cover. Takes any opening with a disturbance date and locates opening with no forest cover recorded against it. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with No SU's

Uses SIL-Opening and ISS-Stocking Standards. Takes SP type openings with a disturbance date and locates openings with no standards unit associated with it. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with SUs with No Early FG Yrs

Uses SIL-Opening and ISS-Stocking Standards. Takes SP type openings with a disturbance date and locates openings with no early free growing years entered within the SU. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with SUs with No Late FG Yrs

Uses SIL-Opening and ISS-Stocking Standards. Takes SP type openings with a disturbance date and locates openings with no late free growing years entered within the SU. Note that this query uses outer joins to look for missing information.

Exception-ISIS Openings with SUs with No Regen Delay

Uses SIL-Opening and ISS-Stocking Standards. Takes SP type openings with a disturbance date and locates openings with no regeneration delay

entered within the SU. Note that this query uses outer joins to look for missing information.

Appendix B

Module/Function examples

Module GreenUpEstimate

Contains two functions GreenupSIHt and GreenupSiteClassHt.

Function GreenupSIHt Description:

Passes two variables, species and site index, to determine the average annual height growth (in metres). This can "grow" the tree height to current year since forest cover information reflects the height growth at a given reference year. The difference between the current year and reference year can be calculated and multiplied by the average annual height growth to provide an estimated current height to then do subsequent analysis. Note that you can copy/paste/modified portions of the function to cover off all species which occurs within the district.

```
Function GreenupSIHt(CurrentSpp As String, SI As Integer) As Double
If CurrentSpp = "Fdi" Then
Select Case SI
   Case 1 To 10
   GreenupSIHt = 0.1
   Case 11 To 20
   GreenupSIHt = 0.2
   Case 21 To 25
   GreenupSIHt = 0.3
   Case 26 To 30
   GreenupSIHt = 0.4
End Select
End If
```

```
If CurrentSpp = "Pli" Then
Module GreenUpEstimate cont.
Select Case SI
   Case 1 To 10
   GreenupSIHt = 0.05
   Case 11 To 20
   GreenupSIHt = 0.1
   Case 21 To 25
   GreenupSIHt = 0.2
   Case 26 To 30
   GreenupSIHt = 0.3
End Select
End If
End Function
```

Function GreenupSiteClassHt Description:

Passes two variables, species and site class, to determine the average annual height growth (in metres). The site class may also be useful since not all forest cover records have a site index associated with it. Some sort of conditional if statement may be used to choose result from GreenupSIHt first, if nothing is available due to blank site index, to use GreenupSiteClassHt instead. This function can be copied/modified/adapted to include all species that occur within the district.

```
Function GreenupSiteClassHt(CurrentSpp As String, SiteClass As String) As
Double
If CurrentSpp = "Fdi" Then
Select Case SiteClass
   Case "L"
   GreenupSiteClassHt = 0.1
   Case "P"
   GreenupSiteClassHt = 0.2
   Case "M"
   GreenupSiteClassHt = 0.3
   Case "G"
   GreenupSiteClassHt = 0.4
End Select
End If
If CurrentSpp = "Pli" Then
Select Case SiteClass
   Case "L"
   GreenupSiteClassHt = 0.05
   Case "P"
   GreenupSiteClassHt = 0.1
   Case "M"
   GreenupSiteClassHt = 0.2
   Case "G"
   GreenupSiteClassHt = 0.3
End Select
End If
End Function
```

Module ConvertSpecies

Function strConvertSpp Description:

Passes tree species and converts the detailed tree species to tree groupings. Note that some species have been removed in this example to save space; however, it is complete species list in the module. This is useful if you are trying to rollup general species categories.

```
Function strConvertSpp(strTreeSpecies As String) As String
Select Case strTreeSpecies
Case "A"
strConvertSpp = "Hardwoods"
Case "AC"
strConvertSpp = "Hardwoods"
Case "ACB"
strConvertSpp = "Hardwoods"
Case "ACT"
strConvertSpp = "Hardwoods"
Case "AT"
strConvertSpp = "Hardwoods"
Case "AX"
strConvertSpp = "Hardwoods"
Case "BA"
strConvertSpp = "Abies"
Case "C"
strConvertSpp = "Cedar"
Case "CW"
strConvertSpp = "Cedar"
...etc...
Case "HW"
strConvertSpp = "Hemlock"
Case "LS"
strConvertSpp = "Larch"
Case "PLI"
strConvertSpp = "Pine"
Case "PW"
strConvertSpp = "Spruce"
Case "T"
strConvertSpp = "Hardwoods"
Case "TW"
Case "YC"
strConvertSpp = "Cedar"
End Select
```

```
End Function
```

Module FS922-ConvertActivityCodes

This module contains four functions, ConvertBase, ConvertTech, ConvertMethod, ConvertDisturb.

Function ConvertBase Description:

Passes one variables, base code, and converts it to a codes and description that is used FS922 for FCAP.

```
Function ConvertBase(Base As String) As String
Select Case Base
Case "DN"
ConvertBase = "Denudation"
Case "SP"
ConvertBase = "Site Prepration"
Case "PL"
ConvertBase = "Planting"
Case "BR"
ConvertBase = "W-Brushing and Weeding"
Case "JS"
ConvertBase = "J-Juvenile Spacing"
Case "FE"
ConvertBase = "F-Fertilization"
Case "PR"
ConvertBase = "P-Pruning"
End Select
End Function
```

Function ConvertTech Description:

Passes one variables, technique code, and converts it to a codes and description that is used FS922 for FCAP.

```
Function ConvertTech(Technique As String) As String
 Select Case Technique
Case "BI"
ConvertTech = "BI-Biological"
Case "BU"
ConvertTech = "B-Burn"
Case "GS"
ConvertTech = "G-Grass Seeding"
Case "CA"
ConvertTech = "C-Chemical"
Case "CG"
ConvertTech = "C-Chemical"
Case "ME"
```

```
ConvertTech = "M-Mechanical"
Case "HV"
ConvertTech = "L-Logging"
End Select
End Function
```

Function ConvertMethod Description:

Passes one variables, method code, and converts it to a codes and description that is used FS922 for FCAP.

```
Function ConvertMethod(Method As String) As String
 Select Case Method
Case "IMCUT"
ConvertMethod = "T=Commerical Thin"
End Select
End Function
```

Function ConvertDisturb Description:

Passes one variables, disturbance code, and converts it to a codes and description that is used FS922 for FCAP.

```
Function ConvertDisturb(Disturbance As String) As String
Select Case Disturbance
Case "B"
ConvertDisturb = "B-Wildfire"
Case "F"
ConvertDisturb = "F-Flood"
Case "L"
ConvertDisturb = "L-Logging"
Case "P"
ConvertDisturb = "I-Insect"
Case "W"
ConvertDisturb = "W-Windfall"
Case "I"
ConvertDisturb = "Slide"
Case "S"
ConvertDisturb = "Salvage"
End Select
End Function
```

Appendix C

Basic Comparison Operators

The following table lists the comparison operators, an example of how they are entered, and the resulting record selection.

Operator	Meaning	Example	Selected if value is...
=	Equal to	="PL"	Equal to "PL"
<	Less than	<6000	Less than 6000
<=	Less than or equal to	<=#1991/01/01#	Dated on or before Jan 1, 1991
>	Greater than	>#1999/12/31#	Dated after Dec 31, 1999
>=	Greater than or equal to	>=1500	More than or equal to 1500
<>	Not equal to	<>"PL"	Not equal to "PL"

Additional Comparison Operators

There are some additional comparison operators which you can use to compare field values with example values. These operators provide more sophisticated matching criteria to extract the desired records.

Some of these additional operators are listed in the table below:

Operator	Meaning	Example	Selected if value is...
Between	Between two values	Between 15 and 20	In the range 15-20
In	In a set of values	In ("PL", "FP", "RP")	Is PL, FL or RP
Not	Not equal to	Not "Y*"	Does not begin with Y
Like	Like *pattern*. The pattern can include wildcards such as * and ?.	Like "Sm*"	Begins with Sm
Is Null	Field is blank	Is Null	Is blank

Building Expressions

The table below lists some of the many Mathematical and String Operators which can be used to build expressions within an Access query. There are also a whole range of functions which are available for more sophisticated expressions. The *Expression Builder* dialog box can be used to assist you in building complex expressions.

Operator	Definition	Example Expression	Resulting Output
+	Add	[*TREES_PLANTED1*]+[*TREES_PLANTED2*]	1200+550=1750
-	Subtract	[*PLND_TRTMNT_AMT*]-[*ACT_TRTMNT_AMT*]	25-20=5
*	Multiply	[COST/HA]*[#Hectares]	60*3=180
/	Divide	[TOTAL_COST]/[#Hectares]	180/3=60
Left	Truncate from left at specified position	Left([*TREE_SPECIES1_CD*],*2*)	PLI=PL
Right	Truncate from right at specified position	Right([*OPENING*],*4*)	82F03000__50=50
Mid	Extract text starting at specified position, and returning specified number of characters	Mid([*ATU_COMPLETION_DT*],6,2)	1995-03-31=03
&	Join two fields (Concatenate)	[BGC_ZONE]&[BGC_SUBZONE]	ESSFwc
Year	Returns the Year of a date argument	Year ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 1995
Month	Returns the Month of a date argument	Month ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 9
Day	Returns the Day of a date argument	Day ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 27

Operator	Definition	Example Expression	Resulting Output
FiscalDt	Returns the fiscal year of a date argument	FiscalDt([ATU_COMPLETION_DT])	ATU Comp Date = 1995-09-27 Result = 1996
Iif	Returns one of two arguments based on an expression	Iif(([REFERENCE_YEAR]=1995),[AVG_AGE],(1995-[REFERENCE_YEAR]))	
Month	Returns the Month of a date argument	Month ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 9
Day	Returns the Day of a date argument	Day ([*ATU_COMPLETION_DT*])	ATU Comp Date = 1995-09-27 Result = 27
FiscalDt	Returns the fiscal year of a date argument	FiscalDt([ATU_COMPLETION_DT])	ATU Comp Date = 1995-09-27 Result = 1996

***Note**: When you build an expression based on fields within the SIA database, you must enclose the field name in square brackets, and match the name exactly as it appears in SIA. Functions or calculations which refer to fields are also case-sensitive.*

Summary Operators

Access provides a number of other *Summary Operators*. The table below lists the most common of them with a description of what they do.

Operator	Definition
Sum	Sums the values based on the right-most group by field
Avg	Averages the values based on the right-most group by field
Min	Returns the smallest value based on the right-most group by field
Max	Returns the largest value based on the right-most group by field
Count	Counts the number of occurrences based on the right-most group by field
StDev	Returns the Standard Deviation based on the right-most group by field
Var	Returns the Variance based on the right-most group by field
Where	Allows you to specify criteria values interactively whenever the query is run. *Where* is used in combination with a message which is typed (enclosed in square brackets) into the criteria cell. When the query is run, a dialog box consisting of this message, and a text box for the user to input the criteria value is displayed.

Appendix D

Toolbars

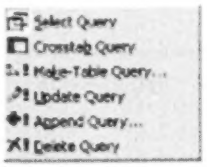
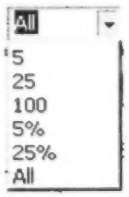
The Datasheet Screen Toolbar

The following table shows the most commonly used buttons in the design screen window.

Button	Purpose
	These buttons are used to switch between Design view, Datasheet view and SQL view respectively (SQL View displays the SQL program created by the query)
	This button saves the query
	Print the currently selected table in the datasheet window
	Display the print preview screen for the currently selected table in the datasheet window
	This button displays the *Find* dialog box which you can use to locate a specific entry within the current field
	Create a new query (part of a drop-down menu)
	Create a new form (part of a drop-down menu)
	Create a new report (part of a drop-down menu)
	This button switches you back to the database window

The Query Design Screen Toolbar

The query design window contains a toolbar with push buttons for common query design tasks. The table below identifies the purpose of each of the buttons on this toolbar.

Button	Purpose
	These buttons are used to switch between Design view, Datasheet view and SQL view respectively (SQL View displays the SQL program created by the query)
	This button saves the query
	The Query Type Button has a drop down menu which allows you to specify either a Select or Crosstab query and Make-Table, Update, Append, or Delete query
	This button runs the query and displays the query result in datasheet view
	This button allows you to add additional tables to your query.
	This button adds a Totals row to the query grid when you want to add summary statistics to your query.
	The Top Values box, finds the top values in the active query based on a percentage of rows or a number of rows. To find the top values, choose a percentage (such as 5%) or enter a number (such as 25).
	This button displays the Properties dialog box for the selected object in the query window
	This button displays the Expression Builder dialog box
	This button switches you back to the database window
	These buttons create a new query, new form or new report based on the current table.

Appendix E

Answer Key for Exercises

Sample Queries (Pg. 48)

Exercise - Modifying Sample Queries - Sorting

16 Perform the sort. Which is the primary sort key? ***Opening***
What order are the sort keys used? ***Plnd_date/ Base/ Tech***

Sample Queries (Pg. 60)

Exercise 2 - Modifying Sample Queries

2 Use the query grid to find out how many records match the following criteria:

a) Treatable Area does not equal 0 (zero) ***1038***

b) Treatable Area does not equal 0 (zero) and
the Maximum Density is greater than or equal to 2500 ***267***

c) Treatable Area is greater than or equal to 40 and
the Maximum Density is greater than or equal to 2500 ***7***

d) Cancel the criteria used in c), then find records for which
the Disturbance date was on or after April 1st, 1993.
Sort the data in descending order of Disturbance Date ***370***

One-Table Queries (Pg. 74)

Exercise - Creating a Parameter Query

8 Use the parameter query to find out how many records match the following criteria:

a) Primary species 1 is "PLI" and Secondary species is "SX" ***142***

b) Primary species is "SX" and Secondary species is "PLI" ***61***

c) Primary species is "SE" and Secondary species is "BL" ***44***

Multi-Table Queries (Pg. 88-89)

Exercise - Inner and Outer Joins

3 Run the query. How many records are included in the output? _ ***731***

7 Run the query. How many records are included in the output? ***3471***

11 Run the query. How many records are included in the output? ***1494***

Advanced Queries (Pg. 110)

Exercise Aggregate Queries

9 What is the total number of trees planted in the 1998 Calendar year? ***751230***

10 What is the total hectares planted in the 1998 Calendar year? ***569.1***

Advanced Queries (Pg. 118)

Optional Exercise – Aggregate Query

What is the average density for trees planted in 1978, 1988 and 1998? ***1288, 1388 and 1300***